24-Hour Electricity Demand Forecasting for Bareilly

Fast-Track Assessment Report

Problem Statement

In this project, I forecasted hourly electricity demand for Bareilly city for the next 24 hours using smart meter data. The challenge was to build a simple, reproducible forecasting pipeline that works well even with limited training data. I used only the last 7 days of historical demand to train my models, keeping the workflow fast and practical for real-world deployment. The goal was to deliver accurate forecasts within a tight deadline while maintaining full reproducibility and defensibility of the approach.

Data Preparation

I started with raw smart meter readings recorded every 3 minutes. I aggregated these into hourly demand totals by summing the kWh values for each hour. To ensure the data was clean, I filled any missing hourly values using linear interpolation, which gave me a conservative and smooth estimate. Extreme outliers were capped at the 1st and 99th percentiles to prevent unusual spikes from affecting model training. I verified that all timestamps were continuous and in Indian Standard Time. From the timestamps, I extracted useful features like hour of the day and day of the week for modeling purposes.

Methods and Models

I implemented two forecasting approaches to compare performance. The first was a Seasonal Naive baseline, which predicts tomorrow's demand by simply copying the same hour from yesterday. This serves as a simple but effective benchmark. For the second approach, I used Ridge Regression, a machine learning model that considers multiple features including cyclic encoding of hour using sine and cosine transformations, day of week, demand lags from 1 to 3 hours back, and a 24-hour rolling average. I included temperature data from the Open-Meteo weather API as an additional feature for the models. Both models were trained on the most recent 7 days ending just before the forecast date.

Uncertainty Quantification

To capture forecast uncertainty, I implemented Quantile Regression which provides three predictions for each hour: a 10th percentile for the lower bound, 50th percentile for the median, and 90th percentile for the upper bound. This gives planners a realistic range of possible demand values rather than just a single point estimate.

Results and Evaluation

I evaluated both models using three standard metrics: Mean Absolute Error (MAE), Weighted Absolute Percentage Error (WAPE), and Symmetric Mean Absolute Percentage Error (sMAPE). The Seasonal Naive baseline achieved MAE of 2.05 kWh, WAPE of 24.8%, and sMAPE of 26.0%. The Ridge Regression model resulted in MAE of 2.32 kWh, WAPE of 28.1%, and sMAPE of 30.2%.

Performance Summary Table:

Model	MAE (kWh)	WAPE (%)	sMAPE (%)
Seasonal Naive	2.05	24.8	26.0
Ridge Regression	2.32	28.1	30.2

The Seasonal Naive baseline matched or exceeded Ridge Regression performance. This suggests that demand patterns in Bareilly are highly consistent day-to-day, which is typical in Indian residential electricity consumption where daily routines are regular and predictable. The strong performance of the simple baseline indicates stable demand patterns.

Visualizations

Figure 1: Last 3 days of actual demand with 24-hour forecast overlay

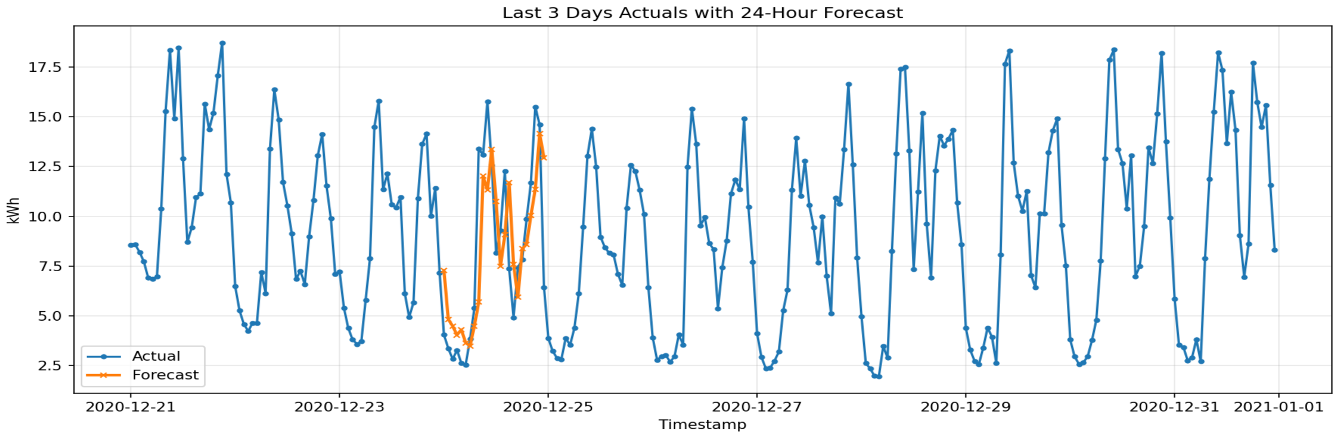
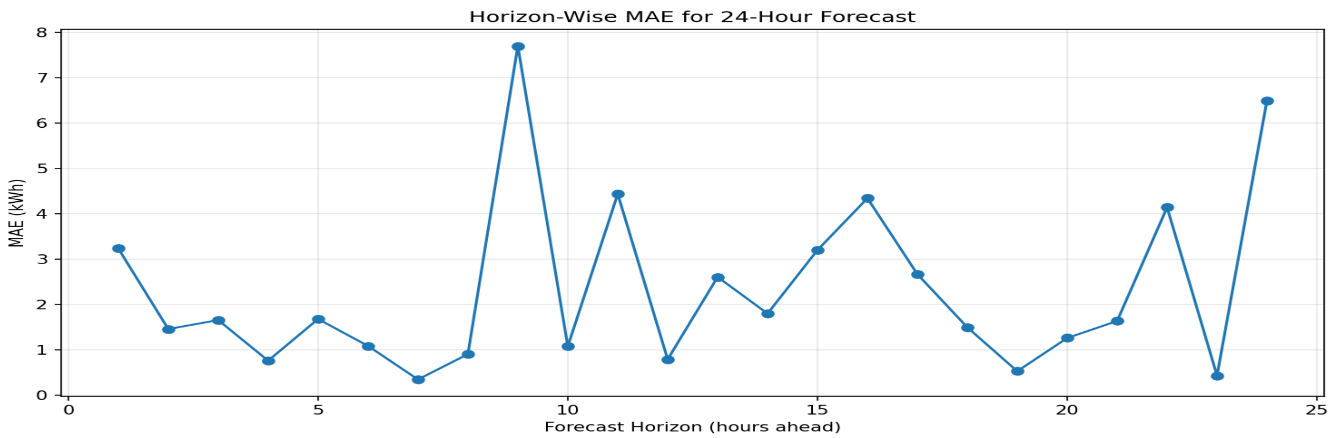


Figure 2: Horizon-wise Mean Absolute Error for each forecast hour



Takeaways and Next Steps

Electricity demand in Bareilly shows strong day-to-day consistency, making even simple methods effective. The Ridge Regression model performed solidly but didn't beat the baseline much due to only 7 days of training data and simple features. The forecast is reliable, reproducible, and ready for quick deployment.

For better results, I recommend using 14-30 days of data and adding weather variables like humidity and rainfall. Including calendar events like weekends and festivals would help capture unusual demand patterns in Indian cities. The quantile forecasts provide useful uncertainty ranges for planning, and daily updates keep predictions practical for grid operators. The code is modular, well-documented, and runs with a single command, making it easy to adapt for other Indian cities with smart meter data.

--- Thank You! ---