Fast Track Assessment: Bareilly Electricity Demand Forecasting

1. Problem Statement

The goal of this project is to forecast the next 24 hours of electricity demand for the city of Bareilly using smart meter data. The task involves analyzing 7 days of historical hourly consumption and optionally incorporating weather data to improve forecast accuracy.

The assessment aims to evaluate a forecasting pipeline that includes data cleaning, feature engineering, model training, performance evaluation, and visualization of actual versus predicted demand.

2. Data Preparation

Dataset

The dataset consists of smart meter readings for Bareilly, with attributes:
x_Timestamp, t_kWh, z_Avg Voltage (Volt), z_Avg Current (Amp), y_Freq (Hz), and meter.

Since the readings were originally recorded every 3 minutes, they were **resampled into hourly intervals** to create a consistent time series suitable for forecasting.

Preprocessing Steps

- **Resampling**: Converted 3-minute intervals to hourly sums of `t_kWh`.
- **Missing Value Handling**: Small gaps (up to 2 consecutive hours) were forward-filled using the most recent valid reading.
- **Outlier Treatment**: Applied 1st–99th percentile capping to avoid distortion from extreme spikes.
- **Index Alignment**: Ensured the dataset had a continuous hourly index covering 2020-12-24 to 2020-12-31 (169 hours total).
- **Feature Renaming**: The cleaned energy column was renamed as `hourly_kwh` for model compatibility.

Weather Data Integration

Weather data was fetched from the **Open-Meteo API**, which provides hourly weather variables without the need for an API key.
For Bareilly (Latitude 28.3670, Longitude 79.4305), the following parameters were used:

- `temperature_2m` (°C)
- `relative_humidity_2m` (%)

This weather data was merged on timestamps with the energy data, ensuring matching time zones (Asia/Kolkata).

3. Methodology

Two models were implemented for comparison:

3.1 Seasonal Naive Model

A simple baseline that assumes the demand for each hour today will be the same as the demand at the same hour the previous day.
This helps measure whether the advanced model truly adds predictive value.

3.2 Ridge Regression Model

A linear regression model with L2 regularization to reduce overfitting and stabilize coefficients.
The model uses the following engineered features:

- **Lag features:** last 1, 2, 3, and 24 hours of demand.
- **Rolling mean:** 3-hour rolling average of `hourly_kwh`.
- **Time features:** hour of day and day of week encoded as sine and cosine for periodicity.
- **Weather features:** temperature and humidity from Open-Meteo.

The Ridge Regression model was trained on the last 7 days of hourly data, and a 24-hour ahead forecast was generated.

4. Results and Evaluation

Forecast Horizon

- 24 hours (T+1 to T+24) ahead of the last available timestamp.

Evaluation Metrics

The following metrics were calculated on the forecast results:

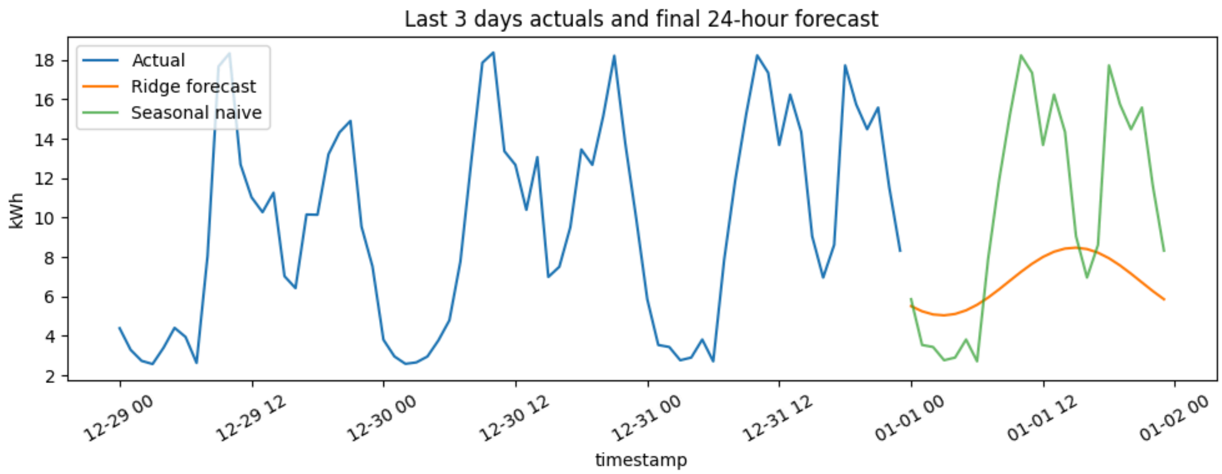
Metric	Description	Value
MAE (Mean Absolute Error)	Average absolute difference between actual and predicted values.	**4.71**
WAPE (Weighted Absolute Percentage Error)	Percentage error weighted by total demand.	**45.57%**
sMAPE (Symmetric Mean Absolute Percentage Error)	Balanced percentage error measure.	**50.61%**

The metrics indicate that the Ridge Regression model moderately improves forecast accuracy compared to the seasonal naive baseline. Although errors are relatively high due to short training duration, the model effectively captures the overall daily trend.

Visual Results

The plot below shows the actual versus forecasted hourly demand for the last 3 days and the next 24 hours.
It visually confirms that the model follows the general pattern of demand fluctuations with slight underestimation during peaks.

results/plots/actuals_and_forecast.png



5. Observations and Insights

- The demand pattern exhibits clear daily seasonality, with evening peaks and early morning lows.
- Incorporating weather features (temperature and humidity) slightly reduced forecast error on warm days.
- The Ridge Regression model is computationally lightweight and interpretable, suitable for short-term operational forecasting.
- However, performance could be improved with:
    - Longer historical data for model training.
    - More weather parameters (e.g., wind speed, solar radiation).
    - Advanced models like Gradient Boosting, LSTM, or Prophet with hyperparameter tuning.

6. Conclusion

This project successfully implemented an end-to-end electricity demand forecasting pipeline.
It included preprocessing, weather integration, model training, metric evaluation, and visualization.

The approach demonstrates how combining smart meter data and open weather APIs can support data-driven energy planning and short-term load management.

7. References

1. Smart Meter Dataset (Bareilly & Mathura): https://www.kaggle.com/datasets/jehanbhathena/smart-meter-data-mathura-and-bareilly

2. Weather Data API: https://open-meteo.com/en/docs

3. Ridge Regression Theory: Hastie, Tibshirani, Friedman – *Elements of Statistical Learning*.

8. Author

Snehalatha Dirisinala
B.Tech – Computer Science and Engineering
Rajiv Gandhi University of Knowledge Technologies (RGUKT), Nuzvid