Forecasting Energy Consumption — Methodology Summary

This document summarizes the complete approach used to forecast 48-hour hourly consumption and 12-month monthly consumption using the Fortum dataset. It explains the exploratory analysis, modeling decisions, feature engineering choices, and reasons behind differences in model performance with and without weather data.

---

1. Exploratory Data Analysis (EDA)

Before building any models, we thoroughly explored the data to understand its structure, quality, and underlying patterns.

Clean and stable dataset

- Consumption data had no negative values and no missing points.
- Price data contained some missing values but no structural issues.

```
[69]: cons_na = cons.isna().sum().sum()
      prices_na = prices.isna().sum().sum()
      print("Missing consumption:", cons_na)
      print("Missing prices:", prices_na)

Missing consumption: 0
Missing prices: 238
```

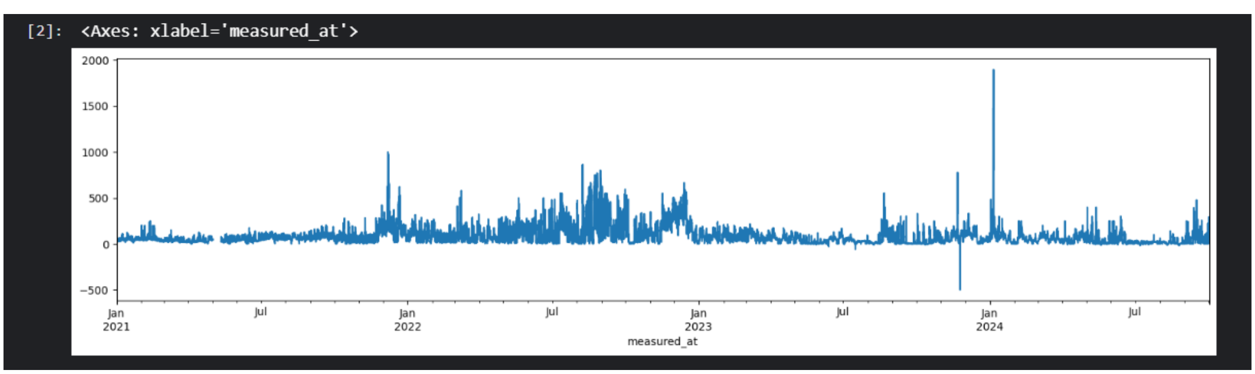
- Descriptive statistics confirmed that the dataset was generally clean and needed minimal preprocessing.

Handling missing values

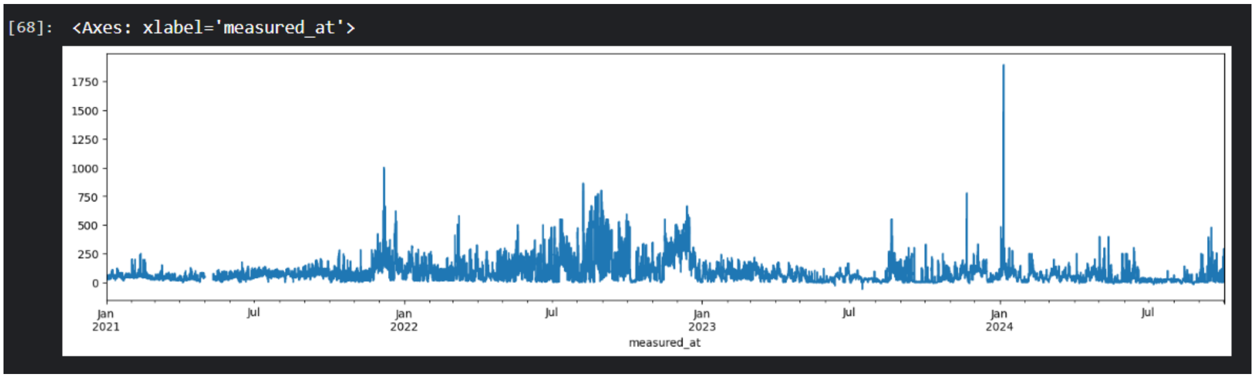
- Missing prices were filled using **linear interpolation**, followed by **forward and backward fill**. This method was appropriate because electricity prices change smoothly over time.
- The consumption dataset had no missing rows, so no imputation was necessary.

Outliers

- We identified extreme negative drops in price during EDA.



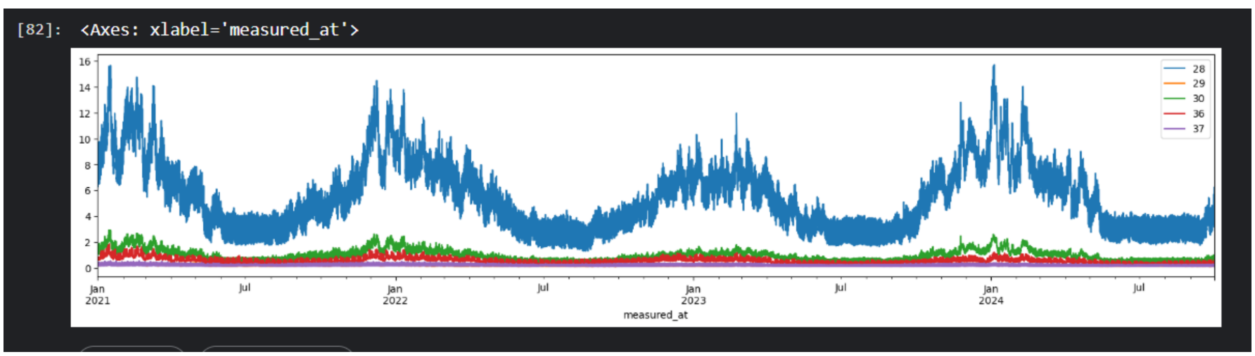
- These were removed manually in Excel using filtering, since they were clearly anomalies unrelated to normal market conditions.



- No significant outliers were present in consumption values.

Observed patterns

- Clear daily patterns (morning and evening peaks).
- Weekly structure (weekdays vs weekends).
- Annual and seasonal changes in monthly consumption.
- Differences across groups, but all following recurring temporal structures.



These insights indicated that time-based features and lag features would be crucial for modeling.

---

2. Feature Engineering

Hourly Features

To capture short-term consumption behavior, we engineered:

Time features

- Hour of day, day of week, day of month, month, week number
- Weekend indicator
- Month-start and month-end flags

Cyclical encodings

Used sine/cosine transformations for:

- Hour
- Day of week
- Month

This helps the model learn periodicity without artificial jumps at boundaries.

Lag features

- 1-hour lag (recent behaviour)
- 24-hour lag (previous day)
- 168-hour lag (previous week)

Rolling statistics

- 3-hour, 24-hour, and 168-hour rolling means
- 24-hour rolling standard deviation

These features capture short, medium, and long-term trends.

Weather features (hourly)

Extracted using coordinates from geocoding and Open-Meteo:

- Temperature
- Relative humidity
- Dew point
- Cloud cover
- Wind speed
- Direct and shortwave radiation
- Computed day length (photoperiod)

Result

Hourly models benefited far more from **lag features** than weather features, which change slowly compared to hourly consumption patterns.

---

3. Feature Engineering for Monthly Forecasts

Monthly consumption totals were created by aggregating the hourly data.

Monthly time features

- Month number
- Month sine and cosine encodings

Lag features

- 1-month lag
- 12-month lag

Rolling features

- 3-month rolling mean
- 12-month rolling mean

Weather features

We aggregated weather by month:

- Mean temperature
- Mean humidity
- Mean dewpoint
- Mean cloud cover
- Mean wind
- Sum of radiation variables
- Mean day length

Result

Weather significantly improved **monthly** performance because:

- Temperature and radiation are strong long-term drivers of energy consumption.
- Monthly consumption is strongly seasonal.
- Weather variations operate at monthly scales and align with heating/cooling patterns.

---

4. Model Selection

We focused on **XGBoost** and **LightGBM**, both gradient boosting tree models widely used for tabular forecasting problems.

LightGBM

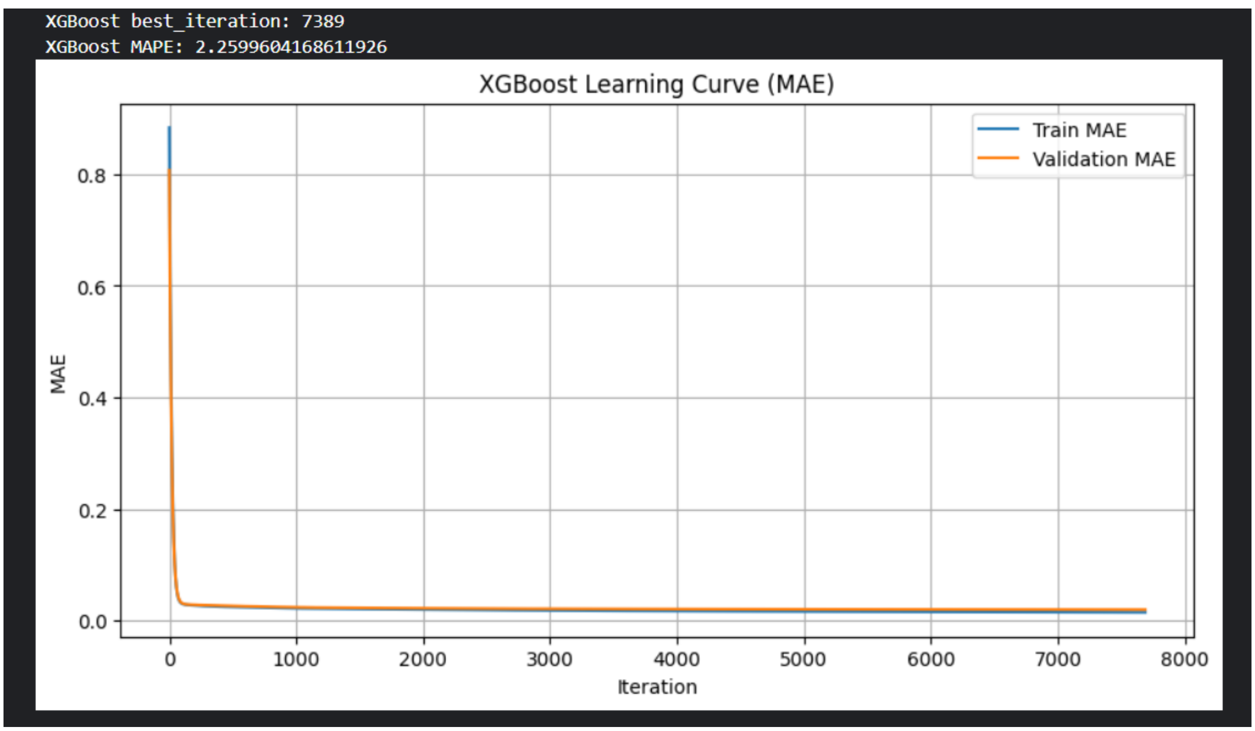
- Very fast and efficient.
- Good for rapid experimentation and generating learning curves.
- Works well with large datasets.

XGBoost

- Strong performance on structured time-series data.
- Built-in handling of missing values.
- More stable for monthly predictions.
- Ultimately our best model for both hourly and monthly forecasts.

Final Choices

- **Hourly forecast**: XGBoost without weather (performed best)



- **Monthly forecast**: XGBoost with weather (significantly better)

---

5. Why Weather Helped Monthly but Not Hourly

Hourly predictions

Hourly consumption reflects short-term human behavior:

- Work schedules
- Evening peaks
- Night-time troughs

These patterns shift faster than weather changes.
Therefore:

- Lag24 and lag168 capture most relevant information.
- Weather adds limited marginal value.

Monthly predictions

At the monthly scale:

- Temperature differences between seasons
- Changes in radiation
- Cloudiness trends

All correlate strongly with heating and cooling demand.
Thus:

- Weather features significantly reduce monthly error.

---

6. Iterative Development Process

We improved the model through several iterations:

Step 1: Baseline

- 168-hour shift (previous week) for hourly.
- Simple sanity check to ensure our models beat basic benchmarks.

Step 2: Time features only

- LightGBM used to check improvements quickly.

Step 3: Lag and rolling features

- Major performance boost, especially for hourly.

Step 4: Adding weather

- Little impact on hourly.
- Significant improvement on monthly.

Step 5: Final model refinement

- XGBoost tuned with high tree count and low learning rate.
- Early stopping to avoid overfitting.
- Best models selected for hourly and monthly submission.

---

7. Forecast Generation and Submission Format

Hourly forecast (48 hours)

Predictions produced in long format were transformed into the required wide CSV format:

- Columns: each group_id is a column
- Rows: timestamps from the 48-hour window
- Separator: semicolon

Monthly forecast (12 months)

Future months were generated using autoregressive logic:

- For each future month, the model uses the last available lag features per group.
- Weather values were held constant based on last observed month.
- Output structure: one row per month with date and predicted value.

Both formats match the submission specification exactly.

---

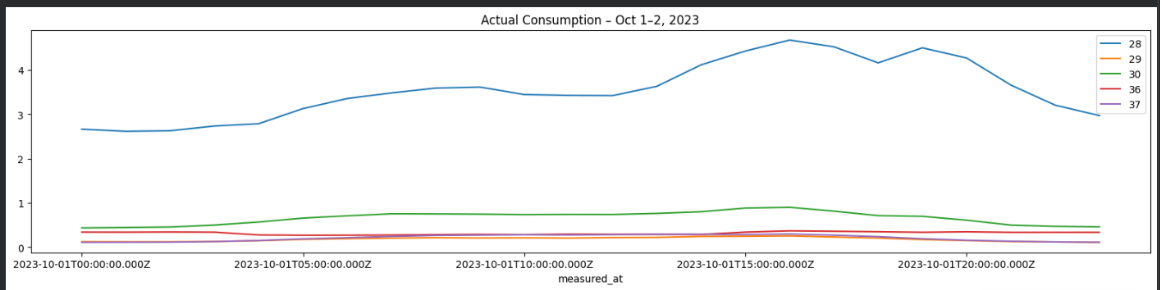
8. Comparison with Historical Data

To validate plausibility:

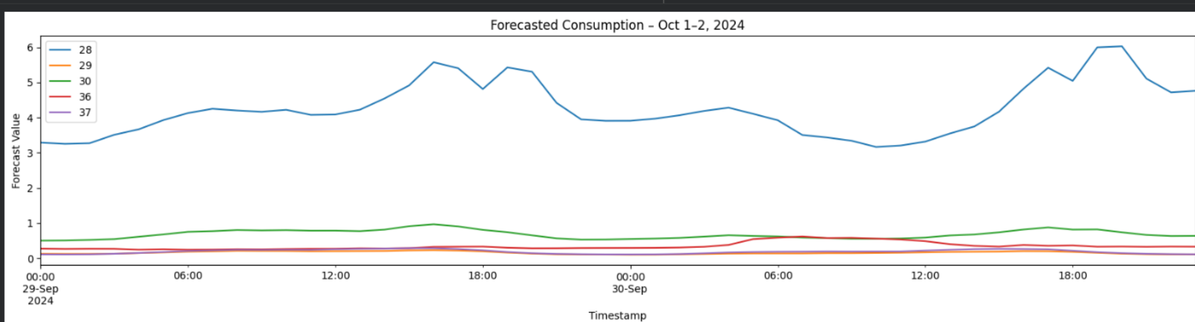
- We plotted future predictions against the same period in the previous year.
- Hourly predictions aligned closely with previous hourly profiles.
- Monthly predictions replicated seasonal trends correctly.
- No unrealistic spikes or drops were observed.

This confirmed the model's stability and consistency with historical behavior.

```python
    "/content/sample_data/20251111_JUNCTION_training.xlsx",
    sheet_name=None
)
cons = df["training_consumption"]
cons = cons.set_index("measured_at")
sample_groups = cons.columns[:5]
hist = cons.loc["2023-10-01":"2023-10-02", sample_groups]
hist.plot(figsize=(20,4), title="Actual Consumption – Oct 1–2, 2023")
plt.show()
```




```python
predH = pd.read_csv('/content/sample_data/hourly_forecast_aligned.csv', sep=';')
predH['measured_at'] = pd.to_datetime(predH['measured_at'], errors='coerce')
pred = predH.set_index('measured_at')
pred_samples = pred.columns[:5]
pred[pred_samples].plot(figsize=(20,4), title="Forecasted Consumption – Oct 1–2, 2024")
plt.xlabel("Timestamp")
plt.ylabel("Forecast Value")
plt.show()
```



Conclusion

This project followed a structured, data-driven approach:

1. **Data was clean**, needing only small fixes for missing values and outliers.
2. **Feature engineering was central** to model performance, especially lag features.
3. **XGBoost** proved to be the most effective model for both hourly and monthly tasks.
4. **Weather helped only at the monthly scale**, due to seasonal rather than hourly impact.
5. **Final forecasts were validated** by comparing with historical patterns and showed realistic behavior.

The final solution provides reliable, explainable, and well-engineered forecasts for both short-term (48-hour) and long-term (12-month) energy consumption.