Electricity Load Forecasting Project — Technical Documnetation/Report

---

1. Project Choice & Real-World Application and Relevance

This project uses the **American Electric Power (AEP) hourly electricity demand dataset**, a real-world time series representing the electricity consumption of a major US utility. The task performed in the provided code—**forecasting future electricity demand**—is a highly practical and valuable data science problem for multiple industries:

- **Energy utilities** need accurate load forecasting to balance supply and demand.
- **Grid operators** use forecasts to schedule generation units and reduce blackout risk.
- **Energy markets** depend on forecasts for pricing, hedging, and purchasing decisions.
- **Renewable integration** relies on good demand predictions to manage variability.

The project builds: 1. A complete data exploration pipeline.
2. Several models (Naive [Baseline], Random Forest, Tuned Random Forest).
3. A realistic forecasting workflow (including a 30-day recursive outlook).

---

2. Dataset and its Description

- **Dataset:** AEP hourly electricity load (MW).
- **Exploratory Data Analysis (EDA) period:** 2015–2016 hourly (17,544 data points).
- **Forecasting period:** 2015–2017 daily averages.

AEP Hourly Electricity Load Dataset

Electric load data is ideal for applied data science assignments because: - It contains clear patterns (daily, weekly, seasonal).
- It shows real-world noise (weather effects, holidays).
- Demand forecasting is a measurable, valuable business problem.

---

3. Data Exploration, Handling & Visualization

DataHandlingnExploration.py}

The provided code performs extensive EDA through multiple plots and transformations. Below is an elaboration—using only what the code does—while adding explanatory depth.

3.1 Patterns in the Hourly Data

The initial line plot of the 2015–2016 data reveals: - **Daily consumption cycles** (night lows, evening highs).
- **Weekly periodicity** (higher use during weekdays).
- **Seasonal variation** (winter + summer peaks).

These patterns arise because: - People use electricity most during mornings/evenings. - Businesses drive weekday demand. - Heating and cooling loads increase consumption during temperature extremes.

Value created:
Recognizing these cycles enables models to leverage lag features and calendar features, improving accuracy with minimal complexity.

3.2 Zoomed January 2015 Plot

Hourly Electricity Load January 2015The zoomed month January 2015 highlights: - Clear 24-hour repeating structure. - Occasional extreme days—likely due to cold spells.

This zoom-in helps validate: - Whether the dataset contains irregularities. - How consistent consumption patterns are.

3.3 Average Hour-of-Day Profile

By grouping by hour of day, the code shows a clear pattern: - Lowest usage around **3–5 AM** - Steady morning ramp - Peak in early evening - Slight midday dip

Average Daily Load HoursOfDay

Insight:
Electricity consumption depends on human routines. This justifies the use of calendar features and lagged daily values.

3.4 Day-of-Week Trends

Grouping by weekday shows: - Monday–Friday: higher demand
- Weekend: lowest demand

Average Load Days-of-Week

This indicates strong industrial/commercial load contributions.

Business interpretation:
Utilities can reduce staffing or maintenance downtime during weekends when load is lowest.

3.5 Monthly & Seasonal Analysis

Monthly average and boxplot visualizations show: - Winter & summer: high load
- Spring & fall: moderate to low

The boxplot reveals wider spreads (more volatility) during extreme temperature seasons, implying: - Weather-driven demand spikes. - Greater forecasting difficulty in peak seasons.

3.6 Load Distribution

The distribution is right-skewed: - Most daily averages fall in the mid-range. - There are rare but impactful high-demand events.

Pitfall:
If a model optimizes only MSE or RMSE, it may under-predict rare extremes—leading to: - Costly energy shortages
- Operational risk

This is why the project includes a **custom asymmetric cost function**.

3.7 Seasonal Trend Loess (STL) Decomposition

The STL decomposition applied to daily averages breaks the load into: 1. **Trend:** multi-month changes
2. **Seasonality:** weekly pattern
3. **Residual:** noise, weather spikes, anomalies

STL is valuable because: - It visually validates model feature choices. - It highlights non-stationarity. - It reveals weekly seasonality clearly.

STL decomposition

---

4. Feature Engineering (As Done in Code)

The code generates a clean supervised dataset with:

Lag-based features

- `lag_1`
- `lag_7`
- `lag_14`

Lag features capture: - Autocorrelation patterns in electricity usage. - Seasonal recurrences (daily/weekly).

Calendar features

- dayofweek
- month
- is_weekend

These reflect behavioral and seasonal influences on consumption.

Why these features?

- They are simple yet powerful for time series like power load.
- They require no external data (weather, holidays, other factors)
- They are fully supported by the patterns revealed in EDA.

---

5. Prediction and Forecasting Models

PredictionnForecasting.py

Three models are implemented and evaluated using: - MAE
- RMSE
- MAPE
- R^2
- Custom cost-aware loss

Below is an elaboration of each.

5.1 Naive (Baseline) (lag-1)

Prediction = yesterday's load.

Metrics from code:

- MAE: 765.08
- RMSE: 991.36
- MAPE: 5.30%
- R^2: 0.6295
- Cost-loss: 1152.28

Interpretation:
The naive model is strong because electricity load is highly autocorrelated, but it fails to incorporate: - Weekly patterns
- Calendar seasonality
- Non-linear relations

It serves as a baseline for all other models.

5.2 Random Forest Regressor

Hyperparameters: - 200 trees
- Depth 10
- Parallelized training (`n_jobs = -1`)

Metrics:

- MAE: 543.15
- RMSE: 716.45
- MAPE: 3.69%
- R^2: 0.8065
- Cost-loss: 807.40

Why Random Forest? - Handles non-linearity. - Robust to noise. - Performs well with mixed feature types (lags + calendar). - Requires no scaling.

This is a major improvement over the naive model.

5.3 Tuned Random Forest

RandomizedSearchCV + TimeSeriesSplit used for hyperparameters.

Best parameters found:

- 200 trees
- Depth 12
- min_samples_split = 10
- min_samples_leaf = 4

Metrics:

- MAE: 539.02
- RMSE: 717.20
- MAPE: 3.66%
- R^2: 0.8061
- Cost-loss: 806.57

Interpretation:
The tuned model provides slight improvements and more stability for future predictions.

---

6. 30-Day Ahead Forecast

30 Days Forecast RandomForest

The code implements: - **Recursive forecasting**
- Each prediction becomes the next day's lag_1
- Features recomputed dynamically

The resulting forecast: - Preserves weekly seasonality
- Smooths noise
- Produces operationally plausible load curves

---

7. Production Considerations

Although not implemented in the code, the logic naturally extends into a production environment.

7.1 Production Pipeline

Data ingestion

- Daily automated pull of hourly readings.
- Resampling to daily average load.

Statistical modelling and Feature engineering

- Compute lag_1, lag_7, lag_14.
- Generate calendar attributes.

Model retraining

- Retrain monthly or quarterly using the latest 2–3 years.
- Validate with time-series cross-validation.

Batch prediction

- Generate next-day forecast daily.
- Produce 30-day rolling forecast weekly.

7.2 Monitoring

Tracking following metrics: - MAE
- MAPE
- Cost-aware loss
- Residual bias
- Seasonal drift

Alert if: - Errors exceed threshold
- Missing data
- Sudden shifts in distribution

Prediction Models Comparison

Model Comparison Across Metrics

7.3 Scaling & Maintenance Considerations

- Deploy the model either via an API for real-time use or as a scheduled batch job.
- Version and manage feature pipelines using tools like Airflow or dbt to ensure consistency.
- Enhance model performance by incorporating weather data as an additional factor.

---

8. Results

Complete results/graphs from the project 1

---

9. Conclusion

- Performs exploratory data analysis using visualizations.

- Builds a structured forecasting pipeline.
- Compares naive (baseline), machine learning, and tuned models.
- Incorporates real-world difficulties and possibilities.
- Provides production-level considerations.

End of Report