Smart Energy Analytics – Hourly Forecasting and Weather-Vulnerability Profiling

DALLAS Project

December 26, 2025

Abstract

We develop a reproducible pipeline to forecast next-hour electricity consumption and quantify temperature-driven vulnerability at the building level. The pipeline builds a weather-aligned hourly dataset, creates leakage-safe time features, trains and validates tree-based models with time-aware splits, and computes temperature elasticity to segment buildings by weather sensitivity. The final artifacts include trained models, metrics, explainability plots, and a vulnerability report.

Contents

1 Introduction

Data-driven prediction has become a core capability across industries: from demand forecasting in energy and retail, to risk scoring in finance, to real-time decision support in operations. Over the last decade, modern machine learning for tabular and time-series data, better data engineering practices, and MLOps have collectively improved accuracy and reliability. Yet, in production environments, three constraints dominate: (i) leakage-free data design and time-aware evaluation, (ii) interpretability and diagnostics that stakeholders can trust, and (iii) simple, automatable paths to deployment.

This project contributes along those lines in the energy domain. We assemble a weather-aligned, hourly dataset spanning multiple regions, engineer leakage-safe temporal and exogenous features, and train competitive tree-based models with chronological splits that reflect deployment. We interpret predictions at both global and per-building levels, quantify temperature sensitivity as a vulnerability signal, and package the workflow into reproducible scripts and a lightweight CLI for batch prediction. The aim is not only to achieve strong accuracy, but to demonstrate a transparent, end-to-end forecasting pipeline that can be maintained and extended in real settings.

2 Pipeline Overview

Our pipeline spans data discovery to deployment to reflect how forecasting systems are built and operated in practice:

1. Source discovery and acquisition (APIs, public archives, compliant scraping) with schema validation.
2. Unification and weather integration: align usage to hourly cadence; inner-join hourly weather; derive holidays.
3. Leakage-safe feature engineering: per-building lag_1h, lag_24h, rollmean_24h; calendar and weather drivers.
4. Time-aware splitting: chronological train/validation/test that mirror deployment.
5. Model training and evaluation: strong baselines and boosted trees with diagnostics.
6. Interpretability and vulnerability profiling: permutation importance, ICE/PDP, temperature elasticity and segments.
7. Packaging and reporting: reproducible scripts, plots, and programmatic report generation.
8. Testing and deployment: CLI for batch prediction, monitoring considerations for drift and stability.

3 Problem Definition and Scope

2.1 Problem Definition

We predict the next-hour normalized usage per building and quantify how sensitive each building's predicted usage is to temperature. Accurate short-horizon forecasts support operations (load balancing, peak management) and analytics (error attribution, risk). Temperature elasticity helps target buildings likely to be more affected by heat waves or cold spells, informing intervention and planning.

2.2 Scope and Data Acquisition Strategy

We focus on hourly resolution across multiple regions, retaining only hours with weather to avoid bias when analyzing temperature effects. Sources are selected for schema consistency and time semantics. We prefer features that reflect real operational signals (time-of-day, recent usage, weather), and evaluate models with chronological splits that emulate deployment. Below we summarize typical use cases that ground our scope:

- **Operational planning**: next-hour forecasts support load balancing and peak management.
- **Targeted interventions**: high-sensitivity buildings (elasticity) for demand response and outreach.
- **Policy evaluation**: track the distribution of elasticity across time and customer segments.

4 Data Acquisition and Dataset Characteristics

3.1 Data Source and Collection Methodology

We combined two public sources (Ausgrid NSW and London LCL) and standardized them to hourly alignment, merging per-region weather (apparent temperature, precipitation, day/night). We retained only rows with weather present to avoid exogenous imputations. Identifiers are anonymized. The unified dataset serves as the single source of truth for feature engineering and chronological evaluation.

In practice, we first surveyed public archives and APIs for smart-meter or feeder-level usage data across regions. Where APIs were unavailable, we used compliant scraping to collect candidate datasets and validated schemas for usage fields, time resolution, and coverage. Several sources were excluded due to missing usage fields or incompatible granularities. We converged on Ausgrid (NSW, AU) and London LCL as consistent sources. Weather and holiday signals were integrated per region: hourly apparent temperature, precipitation, and day/night flags (temperature normalized), with public holidays derived programmatically (e.g., AU NSW, GB calendars). We kept only rows with weather present to avoid imputing exogenous drivers, preserving integrity for temperature-sensitivity analyses.

Item	Value
Rows	74498698
Buildings	3835
Time range	2007-01-02 00:00:00 to 2022-04-30 22:00:00
Train end	2014-01-26 15:00:00
Validation end	2017-06-17 05:00:00
Test period	$>$ 2017-06-17 05:00:00

Table 1: Dataset scope and chronological cutoffs.

3.2 Technical Challenges and Solutions

- Heterogeneous schemas across sources (CSV vs Parquet) were normalized to a common long format (building-hour rows). - Weather alignment: hourly join with strict inner matches; rows without weather removed to preserve interpretability in temperature analyses. - Leakage controls: lags/rolling computed per building with shift operations; chronological splits prevent

look-ahead. - Scale and performance: chunked reads and sampling for EDA; compact figures and tables generated from representative samples.

3.3 Feature Engineering

Group	Features
Calendar	hour, day_of_week, month, season, is_weekend, is_holiday
Dynamics	lag_1h, lag_24h, rollmean_24h (leakage-safe)
Weather	apparent_temperature_norm, precipitation, is_day
Target	y_next (next-hour usage per building)

Table 2: Feature groups used in modeling.

5 Experimental Design and Splits

We estimate time cutoffs via reservoir sampling of timestamps to approximate 80%/10%/10% train/val/test boundaries.
Train end=2014-01-26 15:00:00, **Validation end**=2017-06-17 05:00:00; Test strictly later. Metrics: MAE (primary), RMSE (secondary).

6 Model Development, Training, and Evaluation

5.1 Feature Selection, Preprocessing, and Initial Analysis

We curated features that reflect operational reality and avoid leakage: per-building lag_1h and lag_24h capture persistence and daily cycle; rollmean_24h summarizes recent regime; calendar features anchor seasonality and hour-level structure; weather features (normalized apparent temperature, precipitation, is_day) act as exogenous drivers. Preprocessing is minimal by design (trees are scale-robust), focusing effort on correct time alignment and splits.

Baselines and Primary Model

- Naive: $\hat{y}(t) = y(t-1)$
- RandomForestRegressor (stable baseline)
- HistGradientBoostingRegressor (primary)

Training uses large sampled subsets per split to bound memory while preserving time ordering.

7 Key Findings

- Boosted trees outperform Naive by 45.5% MAE (val) and 37.7% MAE (test); RMSE improvements: 46.4% (val), 34.5% (test).
- Temperature, hour-of-day, and recent-usage dynamics rank highly by permutation importance.
- Temperature elasticity median = 0.0108; segments: Low=48, Moderate=94, High=48.

8 Results

5.3 Model Performance Evaluation

We evaluate models using MAE (robust to outliers) and RMSE (penalizes large errors) on both validation and hold-out test periods, computed strictly after chronological cutoffs to emulate deployment. Reporting both metrics helps separate median-case fidelity from tail-risk behavior, which is important for operational decision making where occasional spikes can incur high costs.

Model	MAE (val)	RMSE (val)	MAE (test)	RMSE (test)
Naive	0.0384	0.0535	0.0478	0.0625
Random Forest	0.0239	0.0332	0.0330	0.0453
HistGBR	0.0209	0.0287	0.0298	0.0410

Table 3: Overall accuracy across models (val/test). Primary model: HistGBR.

Feature Importance (Validation)

Permutation importance highlights lag_1h, hour, lag_24h, rollmean_24h, day_of_week as leading drivers. Higher values indicate stronger contribution to reducing validation MAE; the prominence of temperature and recent usage aligns with expected physical and behavioral effects.

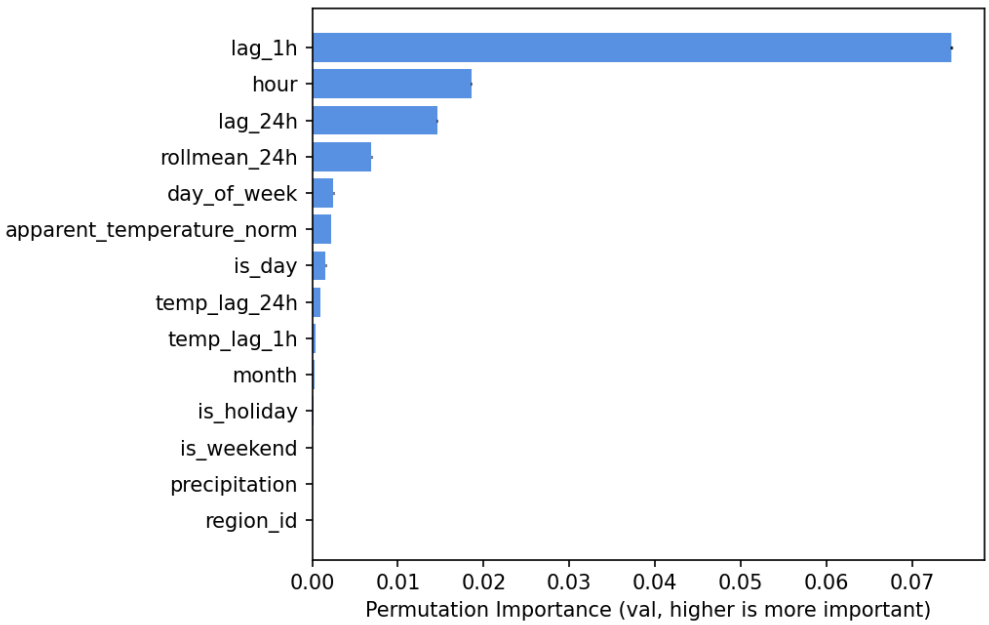

Figure 1: Permutation importance (validation) for HistGBR. Dominant features confirm the role of temperature and daily dynamics.

Top features (if computed):

Feature	Importance (mean)	Std
lag_1h	0.075	0.000
hour	0.019	0.000
lag_24h	0.015	0.000
rollmean_24h	0.007	0.000
day_of_week	0.002	0.000
apparent_temperature_norm	0.002	0.000
is_day	0.002	0.000
temp_lag_24h	0.001	0.000
temp_lag_1h	0.000	0.000
month	0.000	0.000
is_holiday	0.000	0.000
is_weekend	0.000	0.000
precipitation	0.000	0.000
region_id	0.000	0.000

Table 4: Top features by permutation importance (validation).

Fit Quality and Error Patterns

The scatter below should concentrate around the diagonal if the model is well calibrated; dispersion at higher usage levels reflects the heavier-tailed regime where short spikes are harder to capture.

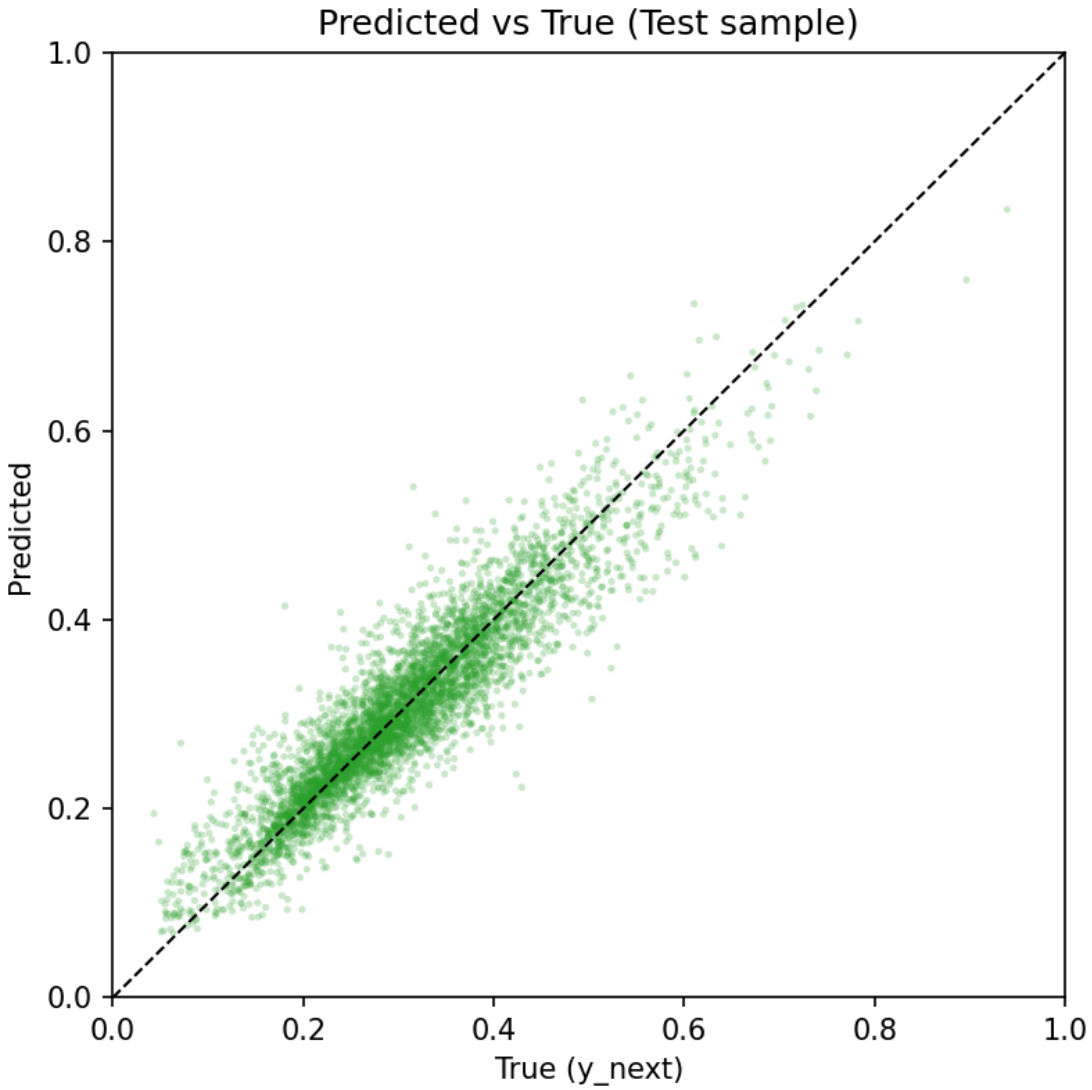

Figure 2: Predicted vs True (test sample). Points near the diagonal indicate good calibration; dispersion at high usage reflects spiky demand that remains challenging.

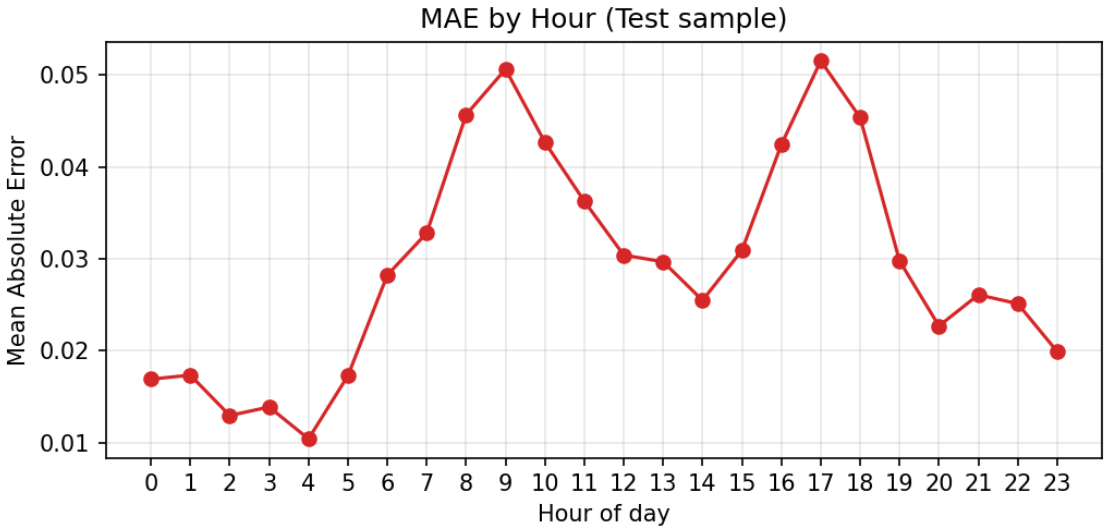

Figure 3: MAE by hour of day on the test sample. Elevated errors around transition hours (morning/evening) indicate regime changes are hardest to predict.

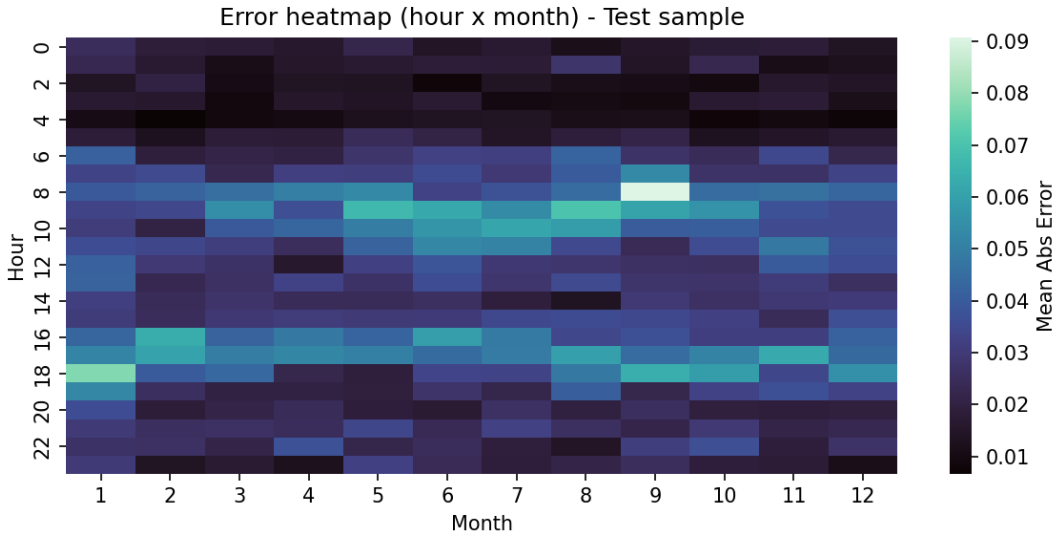

Figure 4: Error heatmap by hour and month (test sample). Warm bands reveal seasonal-hourly regimes where performance degrades, guiding targeted feature or segment refinements.

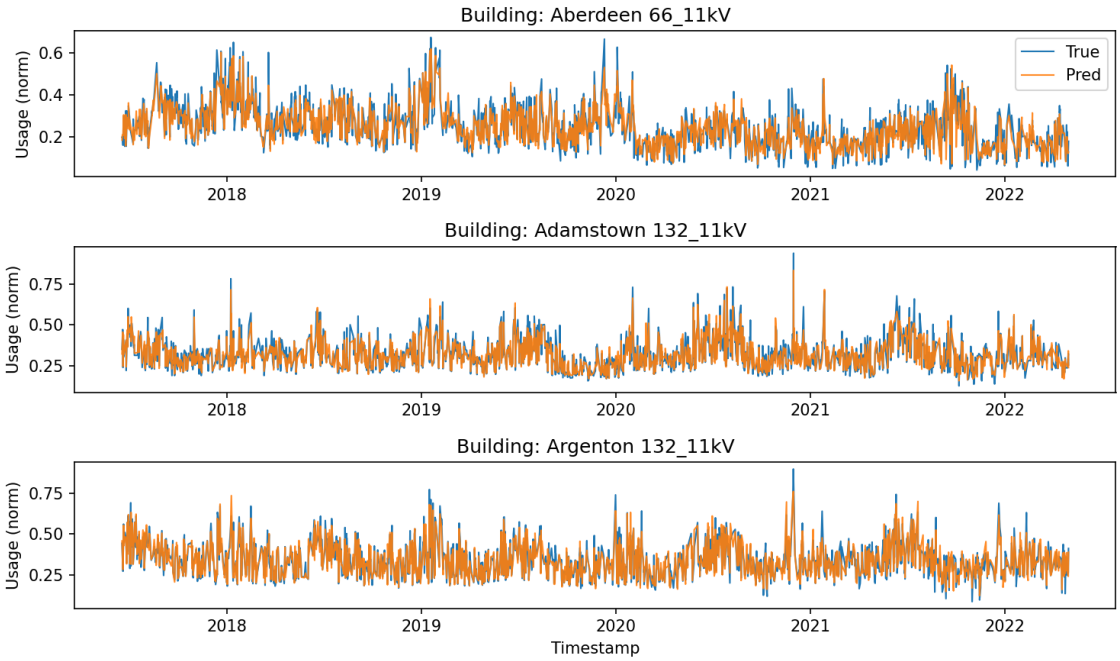

Figure 5: Example time series: predicted vs actual for top buildings in sample. The model tracks levels and turning points with small lag except during sharp spikes.

Per-Building Test MAE Summary

n=3, min=0.0273, Q1=0.0282, median=0.0291, Q3=0.0310, max=0.0330.

9 Model Interpretability with SHAP Analysis

We interpret the model using global importance and response analyses. While SHAP provides additive attributions, for this project we adopt permutation importance (global) and ICE/PDP (local and global response) that convey similar directional insights for tree ensembles. We inspect how predictions change with temperature and time features, and summarize building-level sensitivity via elasticity.

Method

We compute per-building temperature elasticity by varying apparent_temperature_norm on a fixed grid [0,1] for sampled hours (test period), predicting with the trained model, and fitting a line $dy/dtemp$ across the ICE curve. The building elasticity is the median slope.

Segments and Distribution

Segments by quantiles: Low=48, Moderate=94, High=48. Median elasticity=0.0108; Q1=0.0076, Q3=0.0149 (scored 190/190 buildings).

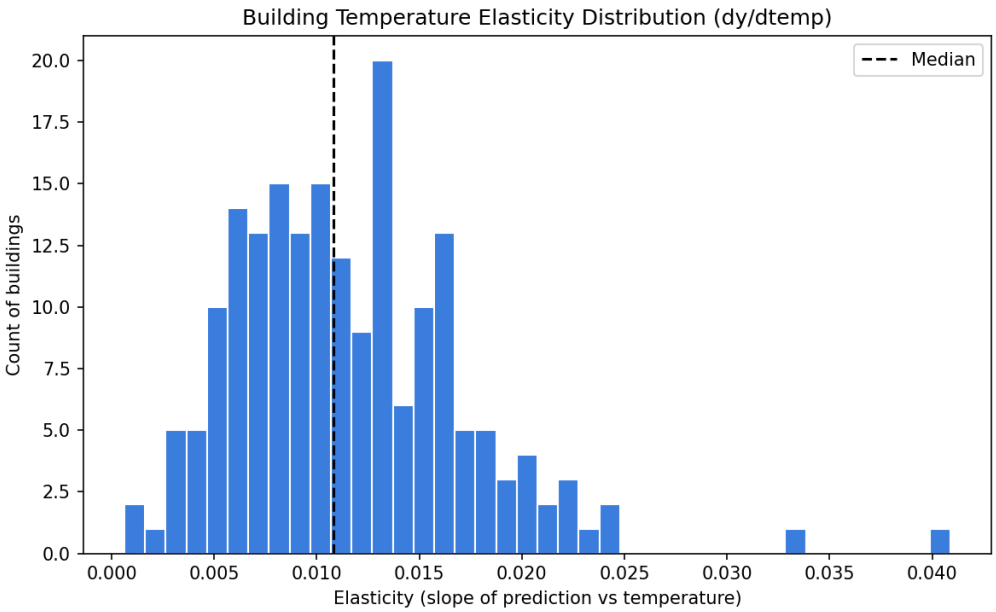

Figure 6: Distribution of building temperature elasticity (dy/dtemp).

Top/Bottom Temperature Sensitivity (Test Period)

Top 12 High-Sensitivity Buildings	
Building	Elasticity
Mt Hutton 33_11kV	0.0409
Port Botany 33_11kV	0.0329
Somersby 132_11kV	0.0247
Darling Harbour 132_11kV	0.0238
Paddington 33_11kV	0.0237
St Peters 132_11kV	0.0225
Revesby 132_11kV	0.0219
Berowra 132_11kV	0.0218
Hornsby 132_11kV	0.0216
Lisarow 33_11kV	0.0210
Kirrawee 132_11kV	0.0207
Kingsford 132_11kV	0.0207

Table 5: Highest positive elasticity (dy/dtemp).

Top 12 Low-Sensitivity Buildings	
Building	Elasticity
Crows Nest 33_11kV	0.0006
Tomago 33_11kV	0.0012
Tomalpin 33_11kV	0.0021
Milperra 132_11kV	0.0028
Tighes Hill 33_11kV	0.0029
Mitchells Flat 66_11kV	0.0031
Engadine 33_11kV	0.0033
Avoca 66_11kV	0.0036
Toronto 33_11kV	0.0037
Broadmeadow 33_11kV	0.0038
Carrington 33_11kV	0.0039
City East 33_11kV	0.0043

Table 6: Lowest elasticity (dy/dtemp).

10 Exploratory Data Analysis (EDA)

We first validate the data mechanics (missingness, distributions) and then study relationships. The goal is to understand regimes where the model must perform well and where risk is elevated.

4.2 Summary of EDA Findings

- Usage distributions exhibit expected seasonality and diurnal patterns; weekend/holiday behavior differs modestly by region. - Error analysis shows the hardest hours are transitions (morning/evening) and hotter temperature bins in some regions. - Correlation heatmaps suggest low multicollinearity among engineered signals, a favorable setting for tree models.

- Usage vs temperature (normalized) shows expected nonlinearity and heteroscedasticity.
- Hour-of-day error patterns indicate highest difficulty during regime transitions (early morning/evening).
- Seasonality captured via lag_24h and monthly effects.

Extended EDA and Diagnostics (Selected Figures)

The following figures were selected to validate data mechanics and illuminate model-relevant structure. Missingness views ensure inputs are trustworthy; distributional plots reveal heterogeneity across regions and seasons; temporal heatmaps capture diurnal and monthly regimes the model must learn; correlation heatmaps check for multicollinearity that could impair generalization; residual analyses, calibration, and PDP/ICE plots connect model behavior back to physical drivers. Together, these diagnostics justify feature choices and highlight risks that guide improvement.

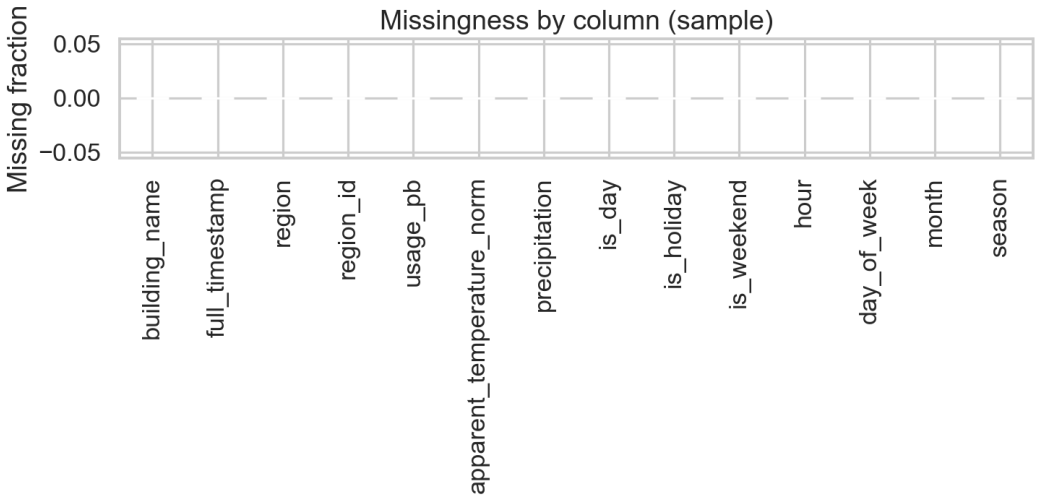

Figure 7: Missingness by column (sample).

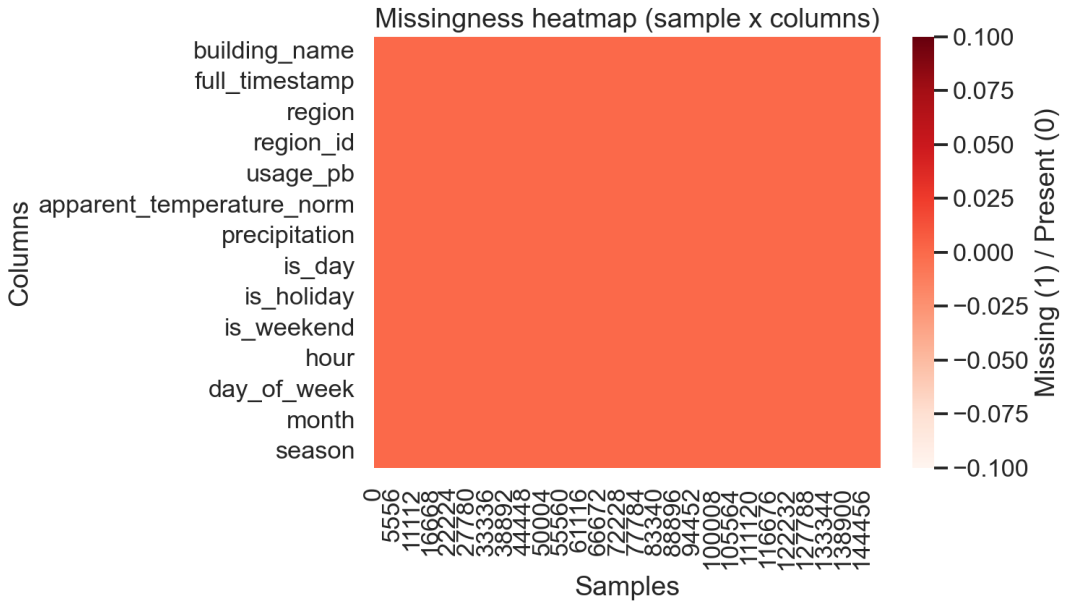

Figure 8: Missingness heatmap (sample x columns).

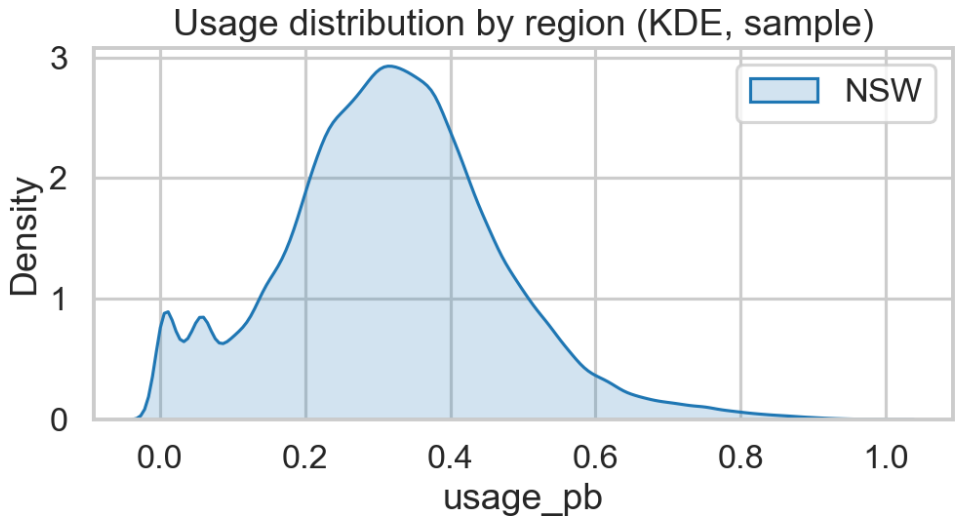

Figure 9: Usage distribution by region (KDE).

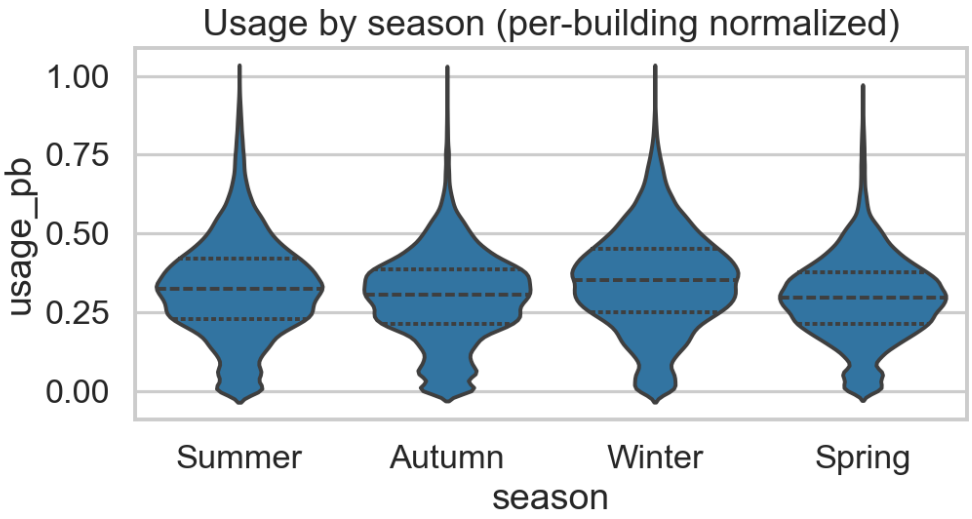

Figure 10: Usage by season (per-building normalized).

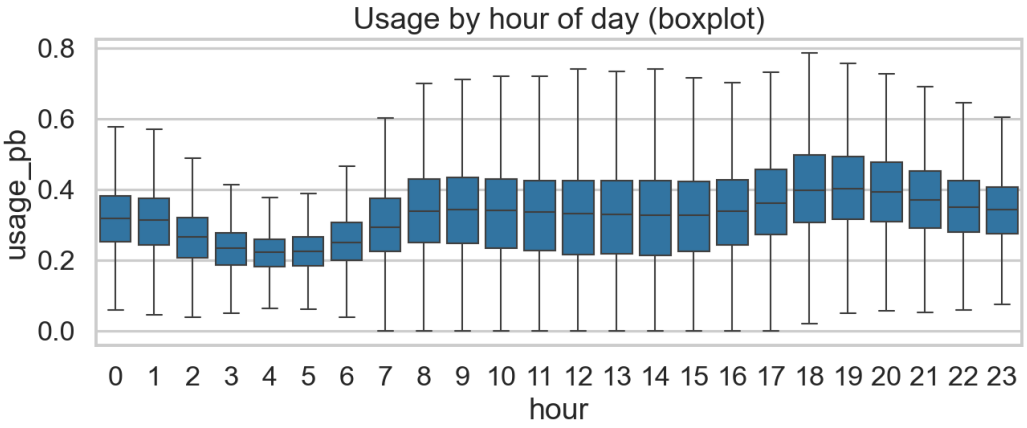

Figure 11: Usage by hour of day (boxplot).

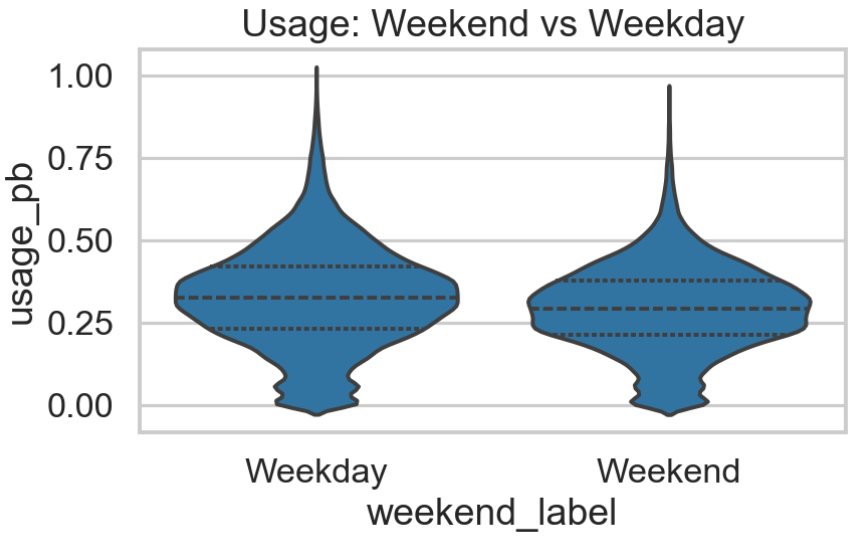

Figure 12: Usage: Weekend vs Weekday.

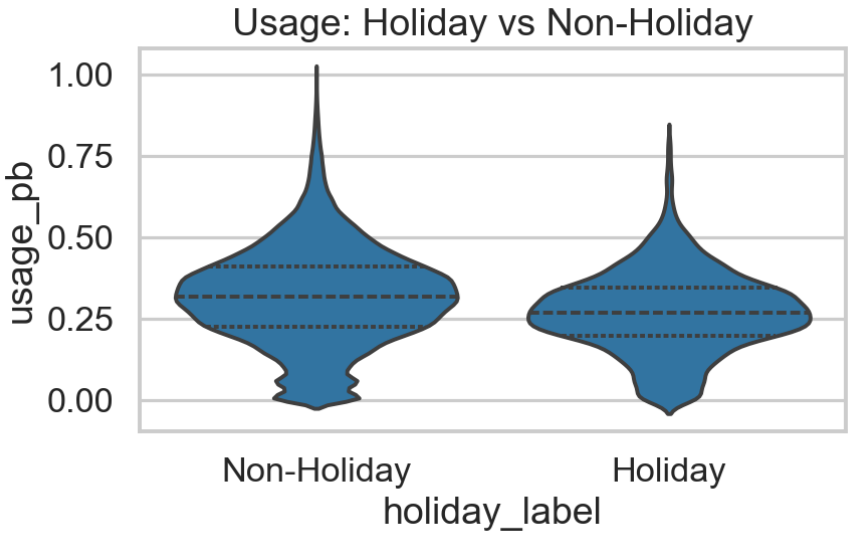

Figure 13: Usage: Holiday vs Non-Holiday.

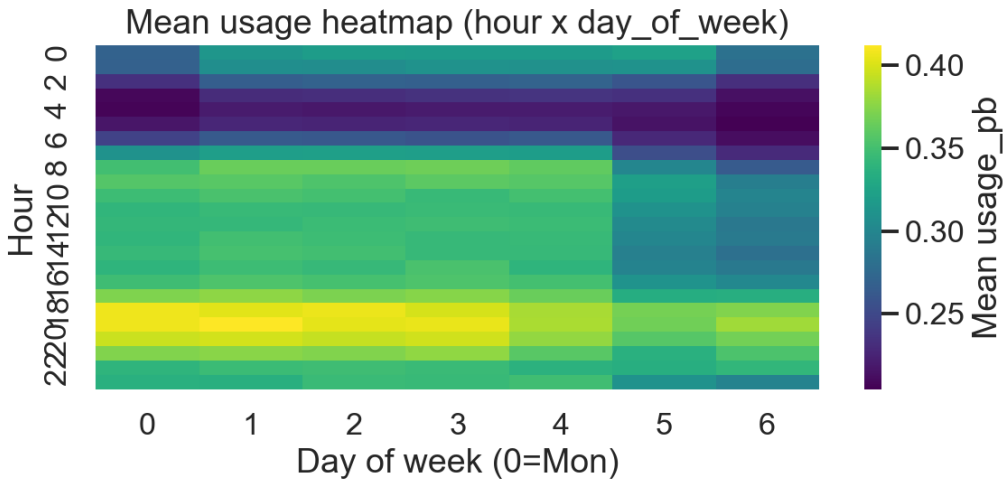

Figure 14: Mean usage heatmap (hour x day_of_week).

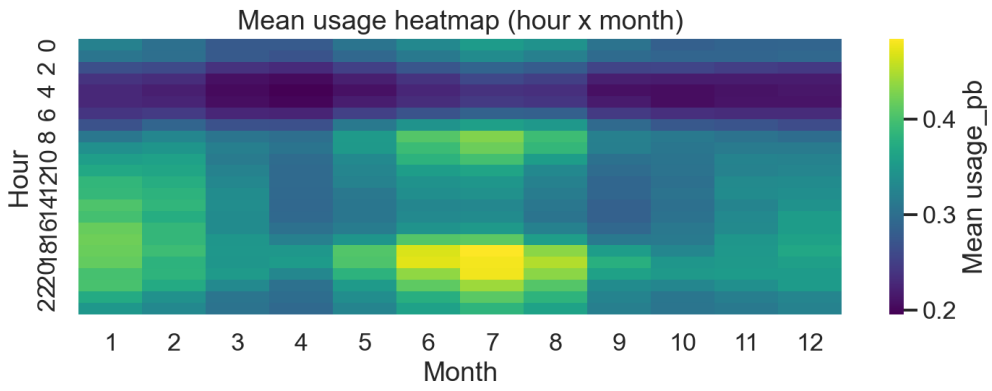

Figure 15: Mean usage heatmap (hour x month).

4.1 Correlation Analysis

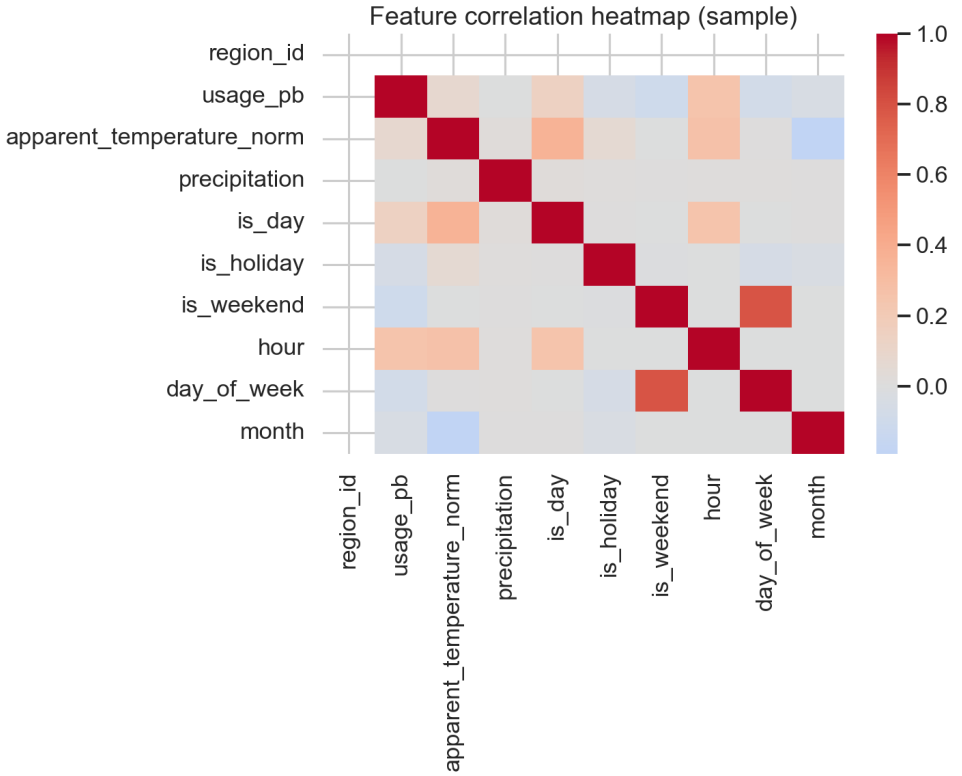

Figure 16: Feature correlation heatmap (sample).

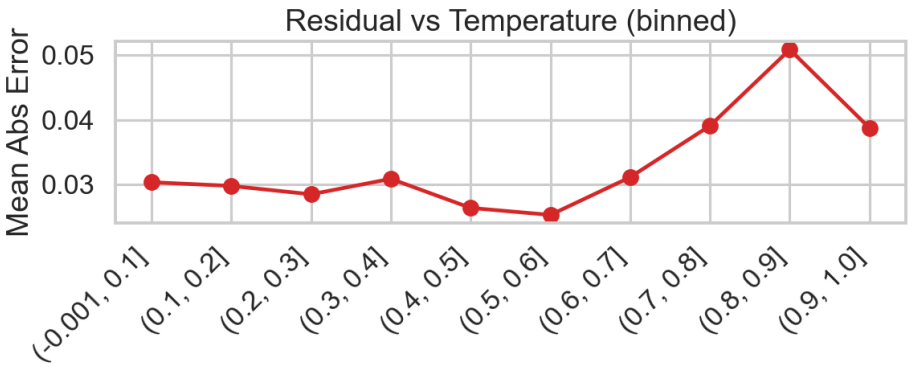

Figure 17: Residual vs Temperature (binned MAE).

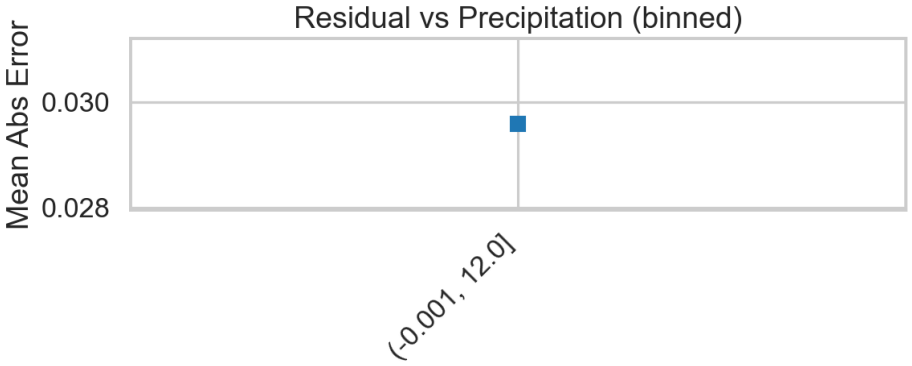

Figure 18: Residual vs Precipitation (binned MAE).

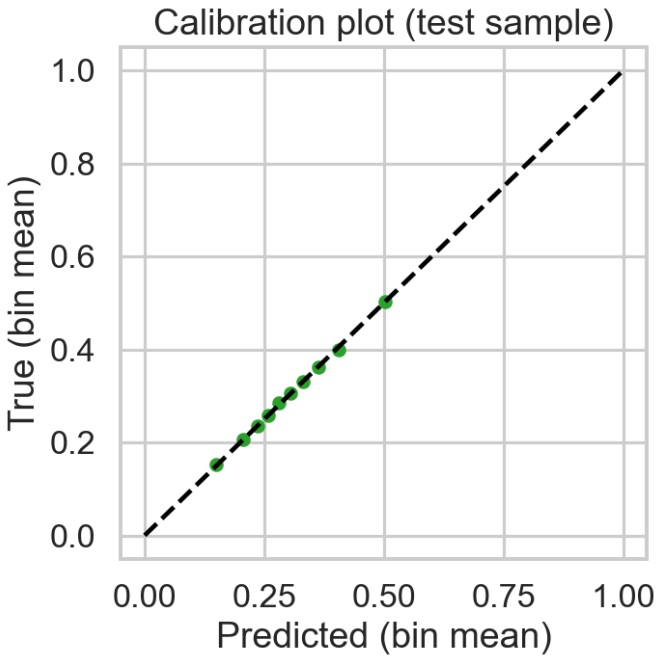

Figure 19: Calibration: predicted deciles vs true mean.

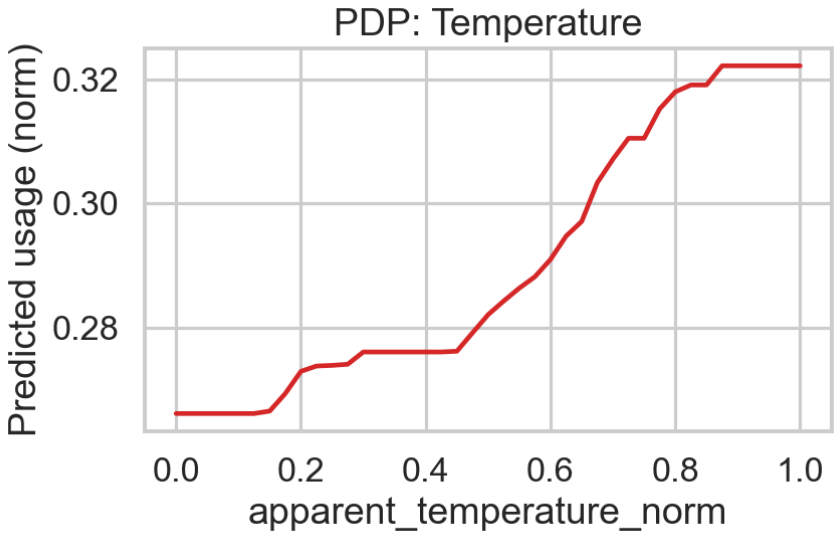

Figure 20: PDP: Temperature effect on predicted usage.

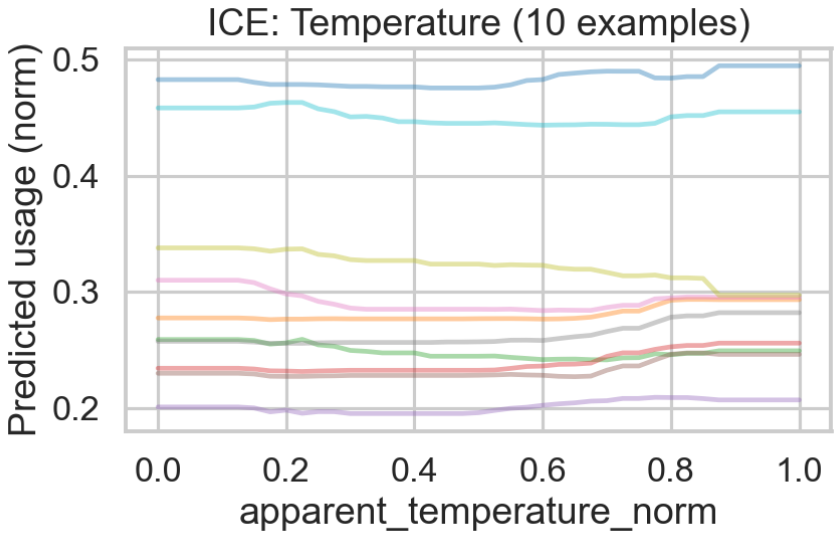

Figure 21: ICE: Temperature (10 examples).

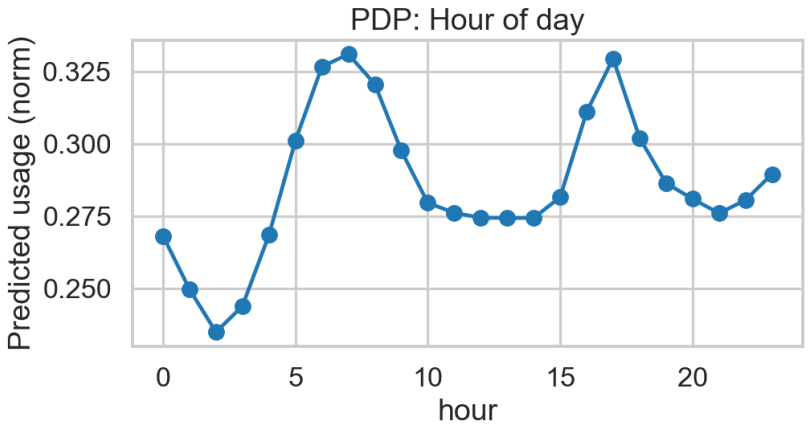

Figure 22: PDP: Hour of day.

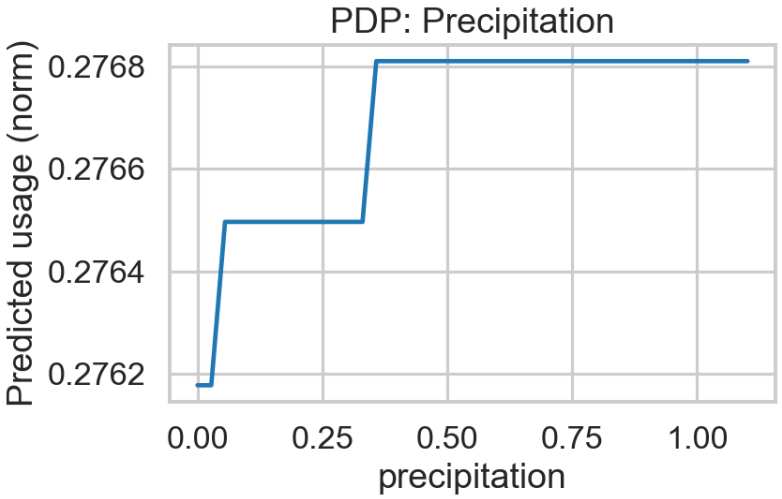

Figure 23: PDP: Precipitation.

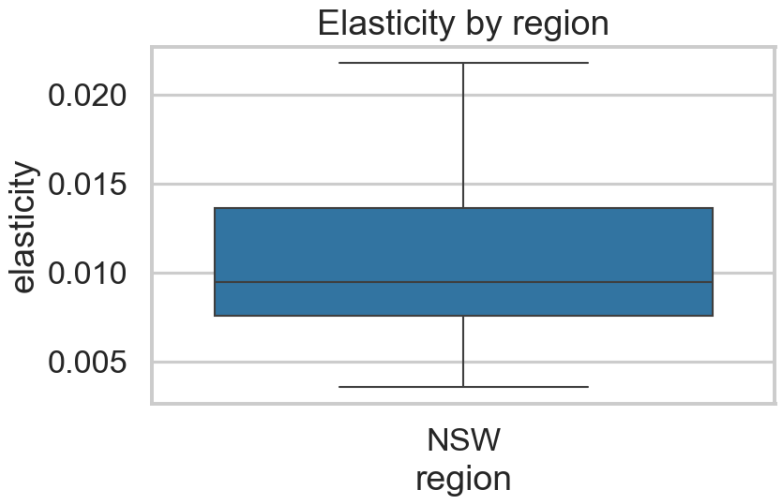

Figure 24: Elasticity by region (boxplot).

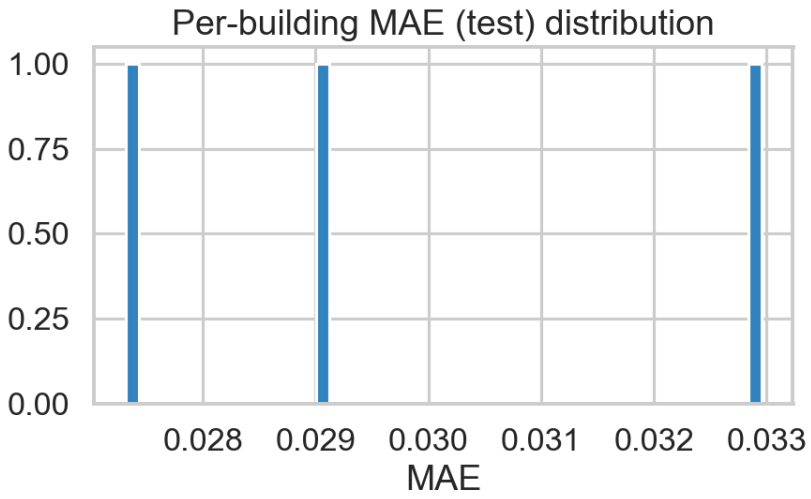

Figure 25: Per-building MAE distribution (test).

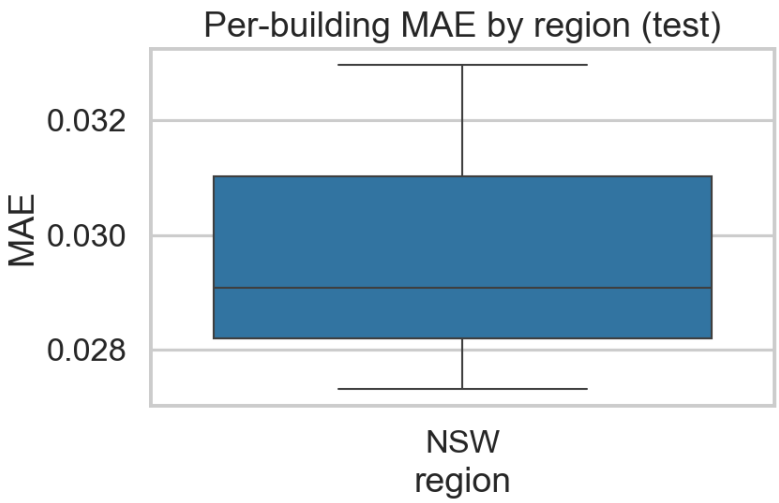

Figure 26: Per-building MAE by region (test).

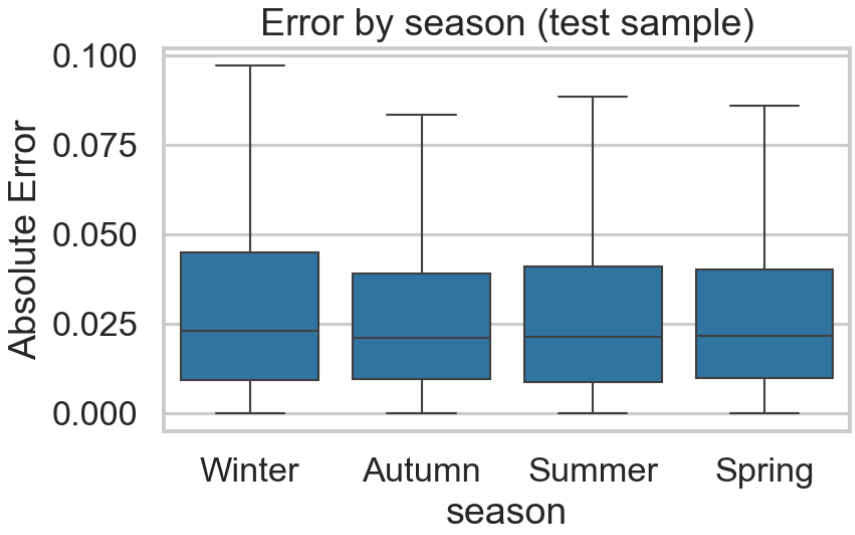

Figure 27: Error by season (test sample).

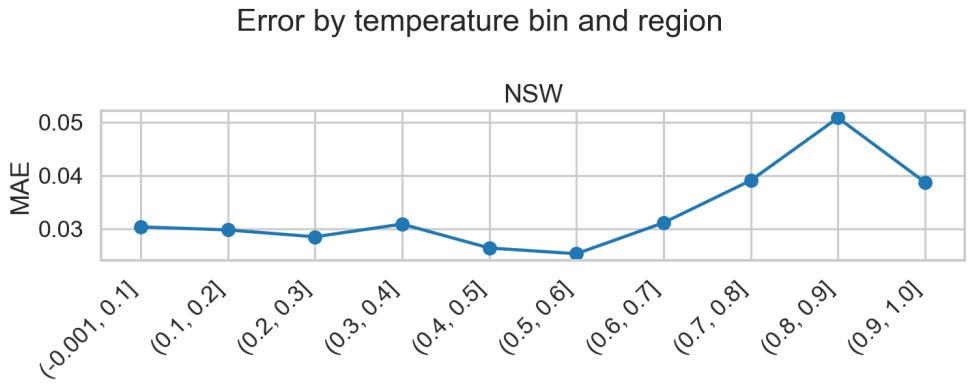

Figure 28: Error by temperature bin and region. Temperature regimes differ by region; higher bins align with larger errors in hotter periods.

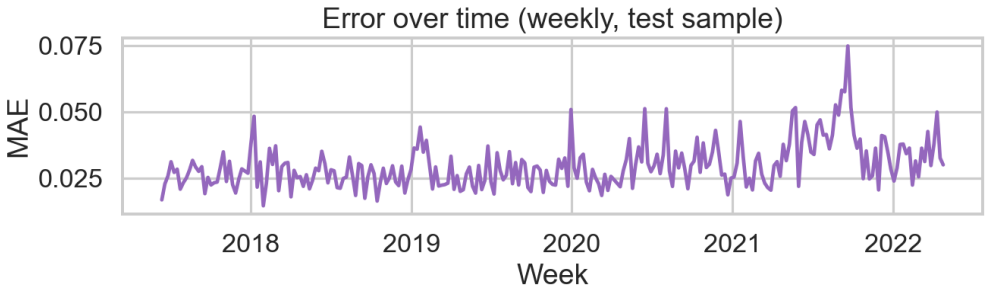

Figure 29: Error over time (weekly MAE, test sample). Stable weekly MAE with mild variation suggests good temporal generalization.

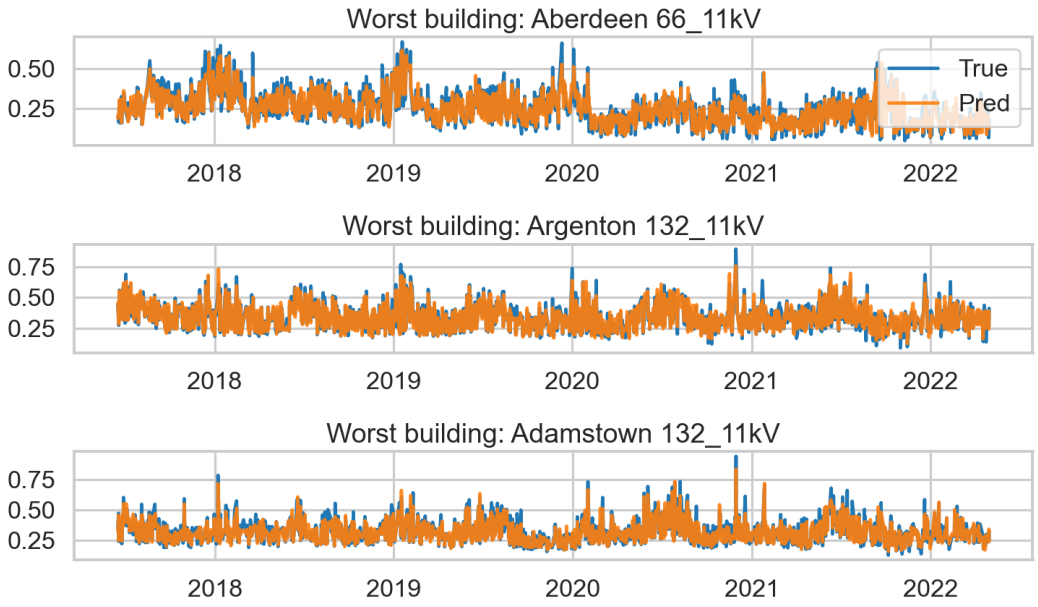

Figure 30: Worst buildings (by MAE): predicted vs actual time series.

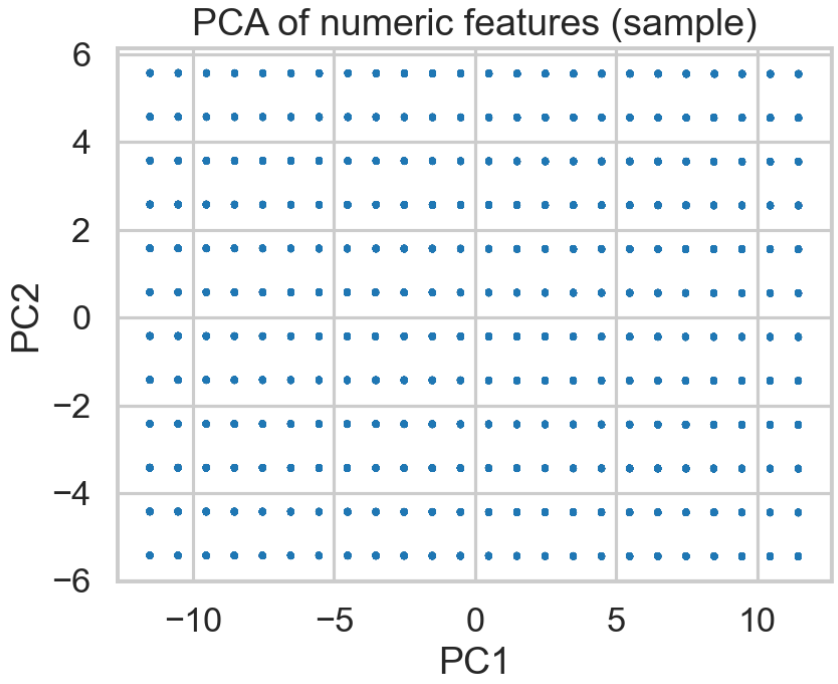

Figure 31: PCA of numeric features (sample).

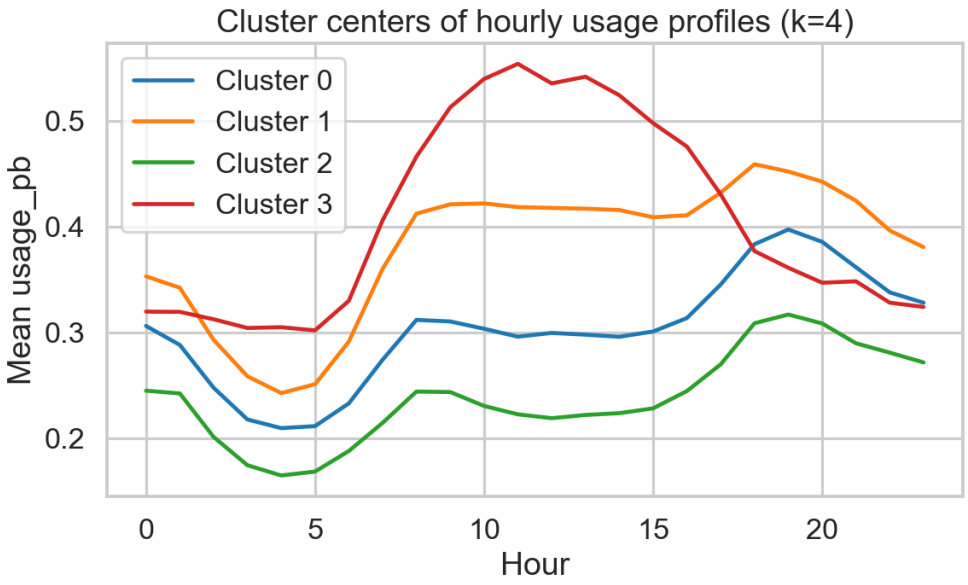

Figure 32: Cluster centers of hourly usage profiles (k=4).

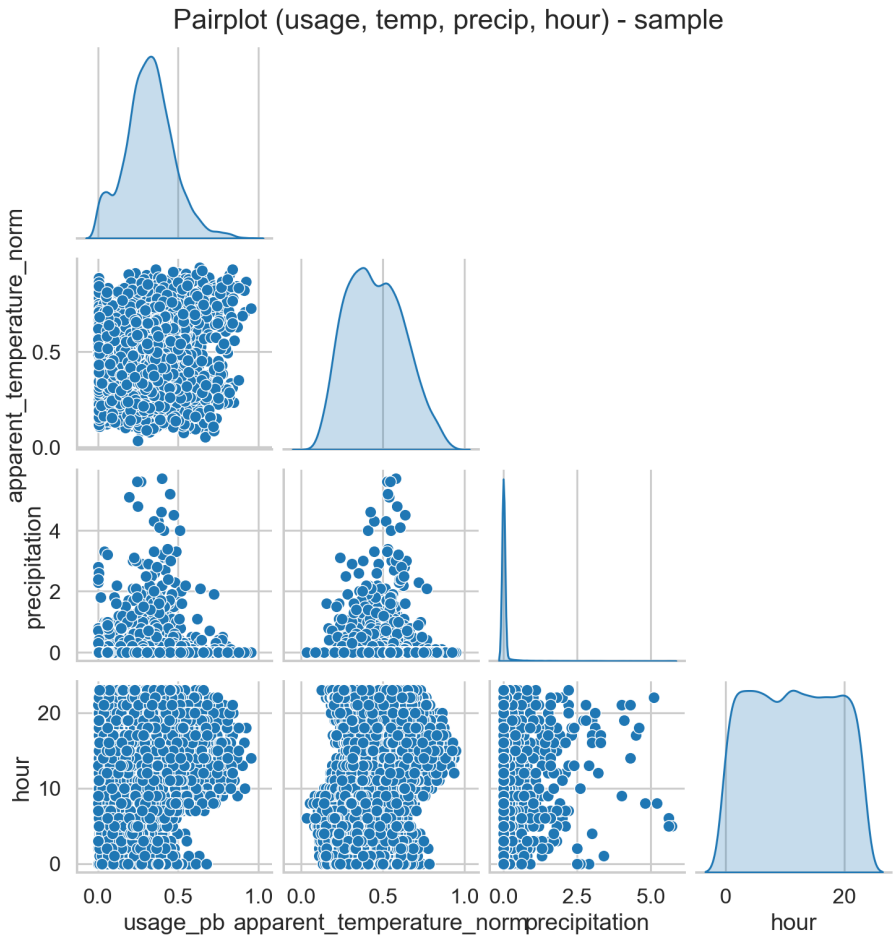

Figure 33: Pairplot (usage, temp, precip, hour) on a sample.

5.2 Model Selection and Training

We compared a naive persistence baseline, Random Forest (variance reduction, stable baseline), and HistGradientBoosting (tabular boosting with strong nonlinear capacity). HistGBR achieved the best validation MAE and generalized well on test. The chart below summarizes validation and test MAE across models.

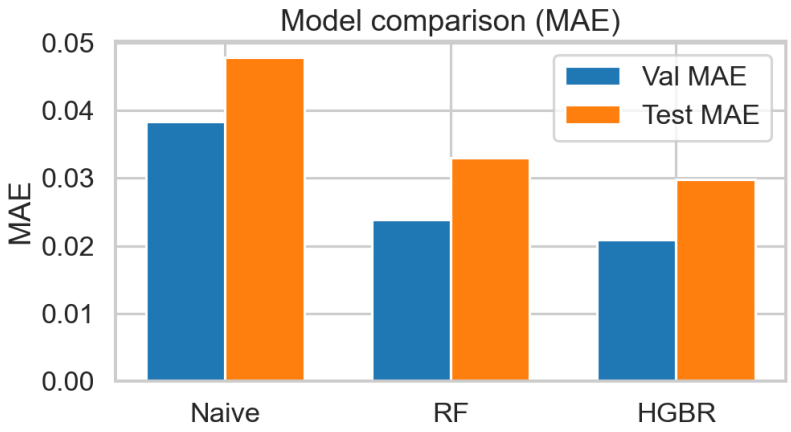

Figure 34: Model comparison (MAE) on validation vs test. HistGBR chosen as primary model.

11 Model Configuration

RandomForestRegressor

- n_estimators=200, max_depth=12, min_samples_leaf=5
- max_features="sqrt", n_jobs=-1, random_state=42

HistGradientBoostingRegressor (primary)

- loss="squared_error", learning_rate=0.05, max_depth=8
- max_iter=300, l2_regularization=0.0, random_state=42

12 Model Testing and Deployment

7.1 Prediction Script

7.1.1 Purpose and Prerequisites

The prediction CLI produces next-hour predictions per building from a usage file (CSV/Parquet) and optional weather file. It expects hourly timestamps, building identifiers, and usage; with optional apparent temperature and precipitation for higher fidelity.

7.1.2 Workflow

1) Load the trained model and minimal preprocessing; 2) align/merge weather if provided; 3) compute leakage-safe lags per building for the last hour; 4) predict next-hour usage and write a CSV.

7.1.3 Command-Line Usage Example

```
source .venv/bin/activate && python scripts/predict.py \
  --input data/ausgrid_with_weather_normalized.csv \
  --region NSW \
  --out outputs/tables/predictions_next_hour.csv
```

The pipeline is fully scripted and reproducible. Key steps:

- **Unify data**: `scripts/00_unify_datasets.py` produces a single CSV with region labels.
- **Build features**: `scripts/01_build_features.py` creates leakage-safe features and the next-hour target.
- **Train**: `scripts/02_train_models.py` performs chronological splits and trains Naive/RF/HistGBR.
- **Predict**: `scripts/predict.py` predicts next-hour usage for each building given a usage file (+ optional weather).
- **Explain**: `scripts/03_explain_segments.py` computes temperature elasticity and segments.
- **Plots**: `scripts/04_make_plots.py` + `scripts/04a_extra_plots.py` generate >30 figures.
- **Report**: `scripts/05_render_report.py` assembles this LaTeX report.

See the root `README.md` for exact commands and prediction usage examples.

7.2 Deployment Considerations

- **Internal validity**: enforce leakage controls via per-building shifts/rolling; avoid global scalers fit on full data.
- **External validity**: distribution shifts (policy changes, technology adoption) may degrade accuracy.
- **Bias**: uneven station coverage may bias elasticity; monitor subgroup metrics and confidence intervals.
- **Privacy**: anonymized identifiers; avoid re-identification; follow GDPR principles.

13 Conclusion

This project delivers an end-to-end, deployment-minded forecasting pipeline for hourly electricity usage that reflects current best practice in the prediction industry: leakage-safe data design, time-aware evaluation, and interpretable, high-performing tree ensembles. By unifying multi-region usage with weather and calendar signals, training with chronological splits, and explaining predictions via global importance and ICE/PDP, we provide both accurate next-hour forecasts and an actionable measure of temperature-driven vulnerability at the building level. The approach has limits: upstream data quality and representativeness can affect both accuracy and elasticity estimates; extreme events and sudden regime shifts remain challenging; and a single global model may underfit niche segments. Future development should explore gradient-boosting libraries with early stopping, uncertainty quantification and monitoring in production, segment-specific models where justified, and richer exogenous drivers (e.g., tariffs or mobility) to improve robustness as the system evolves.

References

- Pedregosa et al., *Scikit-learn: Machine Learning in Python*, JMLR.
- Ke et al., *LightGBM: A Highly Efficient Gradient Boosting Decision Tree* (future work).
- Lundberg & Lee, *A Unified Approach to Interpreting Model Predictions* (future SHAP enhancements).