Appliances Energy Prediction: Nonlinear Regression with Ensemble Methods

Technical Analysis and Performance Evaluation

Author: Mehdi Talebi
Dataset: Appliances Energy Prediction (UCI Machine Learning Repository)

---

1. Introduction

Problem Statement

This project addresses the prediction of household appliance energy consumption (in Wh per 10-minute interval) using environmental sensor data from a low-energy house in Belgium. The dataset, collected over approximately 4.5 months, comprises ~19,700 observations with 25 features including indoor/outdoor temperature, humidity, weather conditions, and time-derived variables.

Why Nonlinear Regression?

Energy consumption exhibits inherently nonlinear patterns driven by occupancy thresholds, temperature comfort zones, and time-of-day effects. Linear models cannot adequately capture these dependencies, motivating the use of nonlinear and ensemble methods that can model complex feature interactions.

Dataset Overview

- **Source:** UCI ML Repository (Candanedo et al., 2017)
- **Observations:** 19,735 (10-minute intervals)
- **Features:** 29 columns (25 predictors + target + date + 2 random noise columns)
- **Target:** `Appliances` – energy consumed by appliances in Wh
- **Missing values:** None

2 Methods

EDA Summary

The target variable is strongly right-skewed (skewness ≈ 3.6), with most readings below 100 Wh and occasional spikes up to 1,080 Wh. Individual feature correlations with the target are weak (max $|r| < 0.3$), suggesting nonlinear dependencies. Clear temporal patterns exist: energy consumption peaks during morning and evening hours, with differences between weekdays and weekends.

Preprocessing

- **Feature engineering:** Extracted `hour` , `day_of_week` , `month` , `is_weekend` from the timestamp
- **Dropped columns:** `date` (after extraction), `rv1` , `rv2` (random noise variables)
- **Outlier treatment:** IQR-based capping on the target variable — outliers were winsorized rather than removed to preserve real high-consumption events
- **Scaling:** StandardScaler fitted on training data only (to prevent leakage), applied to models requiring it (Linear, Polynomial, SVR). Tree-based models used unscaled features.

Models Selected

1. **Linear Regression** – baseline
2. **Ridge Polynomial Regression –** degree 2, α=0.1 (best via CV)
3. **Decision Tree Regressor** – max_depth=20, min_samples_leaf=5
4. **SVR (RBF kernel)** – C=100, γ=0.1, ε=0.5
5. **Random Forest** – 200 trees, min_samples_leaf=2
6. **Gradient Boosting** – 200 estimators, max_depth=7, lr=0.1

Hyperparameter Tuning

GridSearchCV with 5-fold cross-validation (3-fold for SVR due to computational cost) using `neg_mean_squared_error` scoring. Ranges were chosen based on standard practice and dataset characteristics.

3 Results

Model Comparison

Model	Train MAE	Test MAE	Train RMSE	Test RMSE	Train R^2	Test R^2
Random Forest	**7.02**	**14.66**	**10.89**	**22.11**	**0.9359**	**0.7341**
Gradient Boosting	10.70	15.71	14.68	23.16	0.8834	0.7081
SVR (RBF)	12.29	16.16	22.38	26.44	0.7289	0.6196
Decision Tree	9.53	17.13	15.62	27.66	0.8679	0.5837
Linear Regression	26.41	26.42	35.86	35.58	0.3038	0.3111
Polynomial Ridge	26.00	26.62	35.38	36.15	0.3224	0.2889

Key Findings

- **Random Forest** achieves the best test performance (R^2=0.734, MAE=14.66 Wh)
- **Ensemble methods** consistently outperform individual models
- **Nonlinear models** substantially outperform linear approaches (R^2 improvement: 0.31 → 0.73)
- The **Decision Tree** shows significant overfitting (Train R^2=0.87 vs Test R^2=0.58), which Random Forest mitigates through bagging

Feature Importance

The most predictive features across models are: - `lights` (lighting energy) — proxy for occupancy - Indoor temperatures (`T6` , `T3` , `T8`) - Temporal features (`hour` , `day_of_week`)

4 Discussion

Error Analysis

The best model (Random Forest) struggles most with high-consumption events where usage spikes due to simultaneous appliance usage. Large-error cases ($|error| > 2\sigma$) comprise approximately 5% of the test set and are biased toward under-prediction of peak consumption.

Residual analysis reveals: - Near-zero mean residual (unbiased predictions overall) - Some heteroscedasticity: larger residuals at higher predicted values - No strong systematic patterns in residuals vs. predicted values

Feature Interpretation

- `lights` being the top predictor aligns with domain knowledge: lighting correlates with occupancy, which drives appliance usage
- Indoor temperature importance reflects HVAC-related energy consumption
- The `hour` feature captures daily activity patterns (cooking, entertainment)
- Linear model coefficients show positive associations with temperature and lighting, confirming physical intuition

Limitations

- **Single building:** Results are specific to one house in Belgium
- **4.5-month window:** Incomplete seasonal coverage
- **No explicit occupancy data:** Model relies on indirect proxies
- **Independence assumption:** We treat samples as iid, ignoring temporal autocorrelation
- **Feature engineering:** More sophisticated lag features or rolling statistics could improve performance

Future Work

1. **Time-series models** (LSTM, GRU) to exploit temporal dependencies
2. **Real-time occupancy sensors** as additional predictors
3. **Weather forecast integration** for anticipatory energy management
4. **Multi-building generalization** with transfer learning
5. **Automated feature selection** via recursive elimination or LASSO

References

1. Candanedo, L. M., Feldmann, A., & Degemmis, D. (2017). *Data driven prediction models of energy use of appliances in a low-energy house*. Energy and Buildings, 145, 13–25. https://doi.org/10.1016/j.enbuild.2017.03.040
2. UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction
3. Scikit-learn documentation: https://scikit-learn.org
4. Pandas documentation: https://pandas.pydata.org
5. Matplotlib documentation: https://matplotlib.org

AI Usage Statement

Did you use any generative AI tools? No.
All code, analysis, and written content were produced independently.
External resources consulted: scikit-learn documentation, pandas documentation, and the original dataset paper (Candanedo et al., 2017).