Smart Factory Energy Prediction Challenge: Data Preprocessing and Feature Engineering Report

---

1. Data Preprocessing Steps

1.1 Data Loading and Overview

- Loaded a dataset of 16,857 rows and 29 columns from a CSV file.
- Key variables include timestamp, equipment_energy_consumption (target), zone-based temperature and humidity readings, outdoor conditions, and two unlabelled variables: random_variable1 and random_variable2.

1.2 Data Type Conversion

- Converted `timestamp` column to datetime format.
- Cast several object-type numeric columns to proper float format using `pd.to_numeric()` with error coercion.

1.3 Missing Value Handling

- Forward fill (`ffill`) followed by backward fill (`bfill`) used for missing values to preserve temporal trends while ensuring no missing data remains.

1.4 Time Feature Engineering

- Extracted `hour`, `day_of_week`, and `month` from the timestamp for temporal pattern detection.

1.5 Outlier Treatment

- Capped humidity values to the realistic range of [0, 100].
- Clipped wind speed to a non-negative range.

---

2. Feature Engineering and Selection

2.1 Correlation Analysis

- Pearson correlation showed `random_variable1` and `random_variable2` have near-zero correlation with the target:
    - random_variable1: -0.0154
    - random_variable2: -0.0099

2.2 Feature Importance (Random Forest)

- Trained a RandomForestRegressor to assess feature importance.
- Importance scores:

        - random_variable1: 0.0344
        - random_variable2: 0.0305
    - Although low, both variables contribute slightly to the model's predictions.

2.3 Model Performance Comparison

- Compared RMSE with and without the random variables:
    - With: 175.5586
    - Without: 175.7253
- Small performance improvement suggests they have minor utility.

2.4 PCA (Principal Component Analysis)

- Assessed their contribution to the variance structure:
    - PC1 loadings:
        - random_variable1: -0.000161
        - random_variable2: -0.007870
- Conclusion: These variables do not significantly influence the principal components of the dataset.

---

3. Final Decision on `random_variable1` and `random_variable2`

Based on:

- Minimal correlation with the target
- Low feature importance values
- Negligible PCA contributions
- Slight improvement in model RMSE

Conclusion: These variables offer **marginal predictive value**. For production optimization tasks prioritizing interpretability and efficiency, they can be safely excluded. However, if absolute performance is critical, they may be retained.

Tools Used

- Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
- Models: RandomForestRegressor
- Techniques: Correlation, Feature Importance, PCA, RMSE comparison

---

Smart Factory Energy Prediction Report: ML vs DL Comparison

Objective

The aim is to build predictive models that forecast industrial equipment energy consumption based on sensor and environmental data. Two approaches were evaluated:

1. Machine Learning: Gradient Boosting Regressor (GridSearch-tuned)
2. Deep Learning: PyTorch MLP (Fully Connected Neural Network)

🚀 Dataset Summary

- Data Source: Sensor readings from a manufacturing facility
- Features: Zone-specific temperature/humidity, lighting energy, weather data, timestamp
- Target Variable: `equipment_energy_consumption`
- Feature Selection: Top 15 features selected via Recursive Feature Elimination (RF

Model 1: Machine Learning - Gradient Boosting

Model: `GradientBoostingRegressor` from scikit-learn with GridSearchCV

Best Parameters:

```
{
  'learning_rate': 0.05,
  'max_depth': 5,
  'n_estimators': 200,
  'subsample': 1.0
}
```

Evaluation Metrics:

- RMSE: **176.87**
- MAE: **72.68**
- R^2 Score: **0.0821**

Strengths:

- Good baseline for structured tabular data
- Easily tunable and interpretable

Limitations:

- Limited capacity to model complex nonlinearities
- Low R^2 indicates weak generalization

Model 2: Deep Learning - PyTorch MLP

Model: Multilayer Perceptron (2 hidden layers: 128 → 64 units with dropout)

Training Setup:

- Optimizer: Adam
- Loss: MSE
- Epochs: 50
- Batch Size: 64
- Tracked Train vs Test Loss (see attached screenshot)

Evaluation Metrics:

- RMSE: **180.63**
- MAE: **76.76**
- R^2 Score: **0.0426**

Observations:

- DL model underperformed compared to GBM
- Train loss decreases steadily, test loss flattens after ~20 epochs (minor overfitting)

Strengths:

- Flexible architecture for future time-series or sequence-based extensions

Limitations:

- Requires more tuning (learning rate schedulers, early stopping, regularization)
- Lower interpretability compared to tree-based models

Recommendations to Reduce Equipment Energy Consumption

1. **Improve Feature Engineering:** Incorporate shift schedules, equipment types, and operation cycles if available.
2. **Zone-Level Monitoring:** Target high-consumption zones (e.g., high humidity/temp correlation with usage).
3. **Time-Aware Modeling:** Try recurrent models (LSTM/GRU) for capturing temporal dynamics.
4. **Demand Response Strategy:** Schedule equipment use when forecasted demand is low.
5. **Hybrid Models:** Combine ML + DL with ensemble methods or meta-learners.

```
 PyTorch MLP Evaluation:
RMSE: 180.6252
MAE: 76.7623
R² Score: 0.0426
```