Equipment Energy Consumption Prediction Report

Problem Overview

You've been hired as a data scientist for SmartManufacture Inc., a leading industrial automation company. The company has deployed an extensive sensor network throughout one of their client's manufacturing facilities to monitor environmental conditions and energy usage.

The client is concerned about the increasing energy costs associated with their manufacturing equipment. They want to implement a predictive system that can forecast equipment energy consumption based on various environmental factors and sensor readings from different zones of the factory.

Task

Your assignment is to develop a machine learning model that can accurately predict the energy consumption of industrial equipment (equipment_energy_consumption) based on the data collected from the factory's sensor network. This will help the facility managers optimize their operations for energy efficiency and cost reduction.

Specific Goals:

1. Analyze the provided sensor data to identify patterns and relationships between environmental factors and equipment energy consumption
2. Build a robust regression model to predict equipment energy consumption
3. Evaluate the model's performance using appropriate metrics
4. Provide actionable insights and recommendations for reducing energy consumption.

My Solution Walkthrough

1. EDA – Exploratory Data Analysis

- Loaded the dataset using pandas and printed the first 5 rows, checked the info, missing values.
- Explored the Column names and its datatypes.

Insights from these Steps:

1. The dataset contains 16,857 records and 29 columns.
2. Target column: equipment_energy_consumption
3. Timestamps, environmental features (temperature, humidity, pressure, etc.), and zone-specific readings
4. Many features are stored as object (string) types, including numeric values, which need conversion.
5. Some missing values exist across multiple columns.

- Followed by the Description of the target variable
- Converted the non-numerical columns to numerical
- Checked the correlation matrix for the target variable with the remaining feature variables.

Inferences from EDA process

1. Missing Data Top 10 features with missing values: zone1_temperature (949 missing) zone2_temperature, equipment_energy_consumption, zone1_humidity, etc. Target column has 912 missing values and will need careful handling.
2. Target Variable (equipment_energy_consumption) Count: 15,945 (out of 16,857) Mean: 95.8 | Median: 60 | Std. Dev: 182.8 Range: -1139.99 to 1139.99 → likely contains outliers or erroneous data
3. Top Correlated Features with Target lighting_energy: 0.057007 zone2_temperature, zone3_temperature, outdoor_temperature, etc.

The correlations are relatively weak, which may indicate: Complex nonlinear relationships Need for feature interaction terms or model tuning.

2. Data pre-processing

- Found many missing values and imputed them using SimpleImputer with mean.
- Identified and removed outliers using the IQR method to improve model stability and accuracy.
- Visualized distributions and identified multicollinearity risks.
- From visualization found that the feature variables are non-linear to the target variable.

3. Feature Engineering and Selection

Task for Feature Selection

The dataset includes two variables named random_variable1 and random_variable2. Part of your task is to determine, through proper data analysis and feature selection techniques, whether these variables should be included in your model or not. This mimics real-world scenarios where not all available data is necessarily useful for prediction.

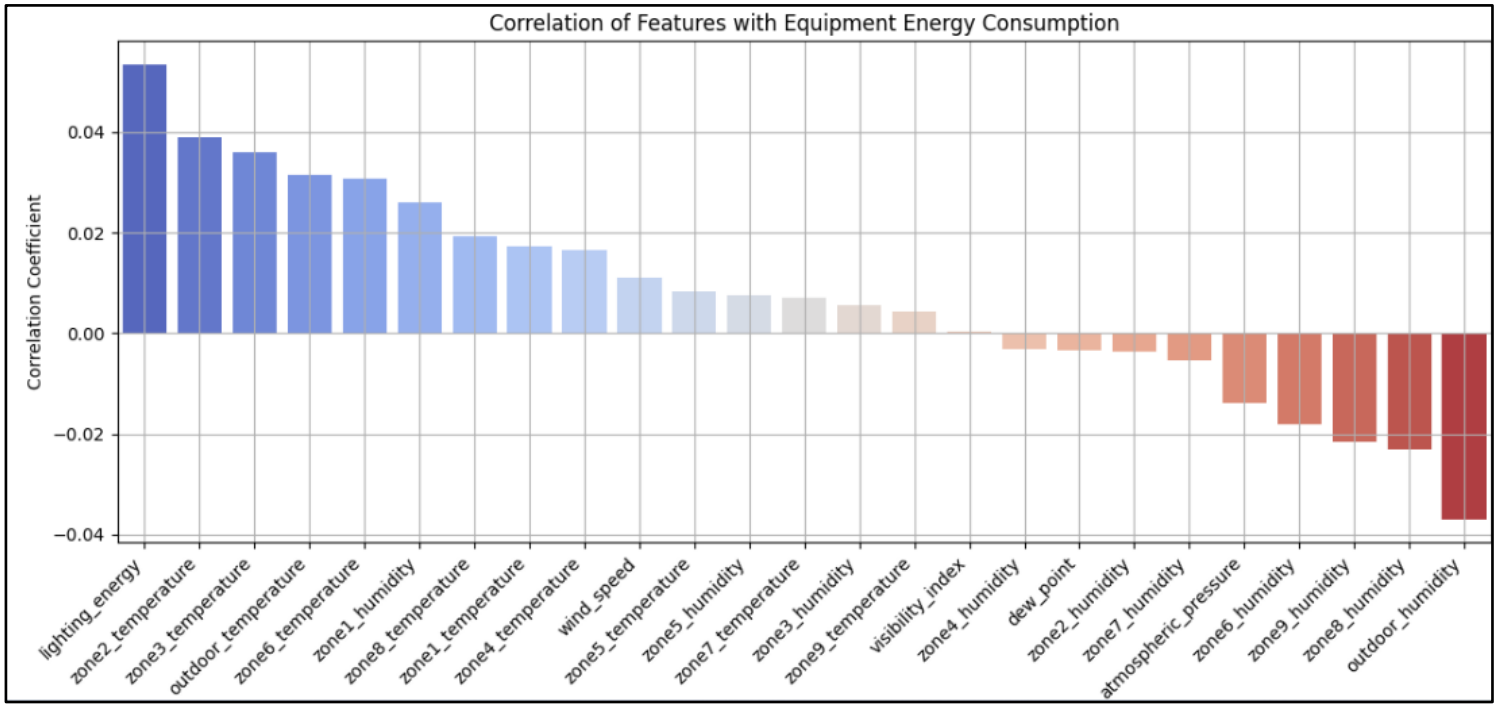
- Analyzed the random_variable1 and random_variable2 using the describe method.
- Correlation of these 2 variables with the target variable equipment_energy_consumption.
- Visualized the distribution of both variables using histogram plot.
- Plotted graph to understand the correlation between these 2 variables.

Inferences from Feature Selection Process

1. random_variable1 and random_variable2 are moderately correlated, suggesting they might be derived from related processes (e.g., same sensor system or formula).
2. But, Both variables have no meaningful correlation with the target variable equipment_energy_consumption.

They likely do not contribute useful signal for energy prediction. Including them in the model might add noise or unnecessary complexity. Hence, Removing both random_variable1 and random_variable2 from the feature set for model training. Dropped the 2 variables from the dataset.

Visualized a bar plot to understand the correlation of features with the target variables



4. Model Evaluation and Development

- I seperated the data into X and y where, X contains feature variables and y contains the target variable.
- Then, I split the X and y into train and test datasets for further model prediction. Now, the data is ready to model development.
- As the data is non-linear I ruled out Linear Regression from the picture. From analysis came to the conclusion that the data is highly complex and cannot work with basic models. Have to use models with complex solutions.
- I first tried Ensemble Learning using **Random Forest Regressor**, the reason for choosing this model is it handles non-linearities and interactions well; robust to outliers.
- Evaluated the model performance for our data,

  MAE: 72.23218

  RMSE = 166.07288

  R^2 Score = 0.03471

  From the rmse and r2 score, we can imply that the RandomForestRegressor model is not that suitable for our data.

- So, I tried **Support Vector Regression (SVR)** as our data has a more complex relationship, SVR with a non-linear kernel (e.g., RBF).

  MAE: 90.91

  RMSE: 34675.56

  R^2 Score: -0.214

- I also tried with multiple models (Linear Regression, MLPRegressor, SVR, Random Forest Regressor)at once and saved the evaluation parameters in a data frame as follows:

	MAE	RMSE	R^2
Linear Regression	75.162839	28322.324850	0.008742
SVR	64.118372	28083.667865	0.017095
Random Forest	71.239310	27338.509991	0.043175
MLPRegressor	130.192345	48011.880240	-0.680376

- Hyperparameter Tuning Approach
  I learnt about GridSearchCV and RandomizedSearchCV , after more research found that RandomizedSearchCV is preferred over GridSearchCV for high-dimensional hyperparameter spaces to save compute time.
  From this approach, came to decsion to use XGBRegressor
  - XGBRegressor handles **non-linear** relationships well.
  - Automatically manages **missing values**.
  - Very powerful for **tabular regression tasks**.

    MAE: 71.79
    RMSE: 165.23
    R^2 Score: 0.045

Interpretation

- **MAE and RMSE** have improved slightly compared to earlier models (e.g., SVR or default TPOT).
- **R^2 Score = 0.045**: This is **still low**, suggesting that the model explains only ~4.5% of the variance in energy consumption. The dataset might:
  - Lack strong predictive features,
  - Contain noise/outliers,
  - Or need further feature engineering.
- Used SHAP to understand *how* each feature affects each prediction

5. Key Findings and Insights

1. XGBoost Performed Best Overall - Among various models (Linear Regression, SVR, MLPRegressor, RandomForestRegressor) **XGBRegressor** consistently achieved the lowest MAE and RMSE and a slightly positive R^2 score.

- Final XGBoost metrics:
  - **MAE**: ~71.79

- **RMSE**: ~165.23
- **R^2 Score**: ~0.045

2. Data Had Weak Predictive Power - R^2 scores across models were low or negative, indicating:

- High noise in the data
- Possibly irrelevant or missing features