Environmental Impact on Energy Consumption: A Data-Driven Approach for Manufacturing Efficiency

Problem Statement

1. What is the problem?

SmartManufacture Inc., an industrial automation company, has installed sensors across one of their client's manufacturing plants. These sensors continuously monitor environmental conditions and track energy usage throughout the facility. Recently, the client raised concerns about their growing energy bills and wants to understand what is driving the high consumption. They are looking for a data-driven solution that can help them forecast energy usage and identify areas of improvement.

2. Why are we analysing energy consumption?

By analysing energy consumption in relation to environmental factors, we aim to uncover patterns and insights that explain when and where energy usage is highest. Understanding these patterns allows us to detect inefficiencies, predict future usage, and ultimately help reduce unnecessary energy costs.

3. Importance of the analysis

This analysis is important because:

- It helps reduce operational costs by identifying energy-saving opportunities.
- It supports predictive maintenance and scheduling of heavy equipment.
- It improves overall energy efficiency in the factory.
- It helps the company move toward sustainable and optimized manufacturing operations

Environment Information

- **Python version**: 3.10
- **IDE used**: Jupyter Notebook
- **Key Libraries**:
  - pandas – for data manipulation
  - numpy – for numerical operations
  - matplotlib & seaborn – for data visualization
  - sklearn – for machine learning and modeling

How to Run the Code

To run this project locally on your machine, follow the steps below:

1. Clone or download the project repository.
2. Open the folder in **Jupyter Notebook**.
3. Run the notebook files in the following order:
   - EDA_Preprocessing.ipynb
   - Feature_Selection.ipynb
   - Build_Evaluate_Model.ipynb
4. The final trained model is saved as ‘*best_rf_model.joblib*’.

Folder Structure

DS-Intern-Assignment--Anshika-Srivastava

1. data

   a. data.csv
   b.cleaned_dataset.csv
   c. processed_dataset.csv

2. docs

   a. data description

3. notebooks

   a. EDA_Preprocessing
   b. Feature_Selection
   c. Build_Evaluate_Model
   d. best_rf model.joblib

4.ReadMe

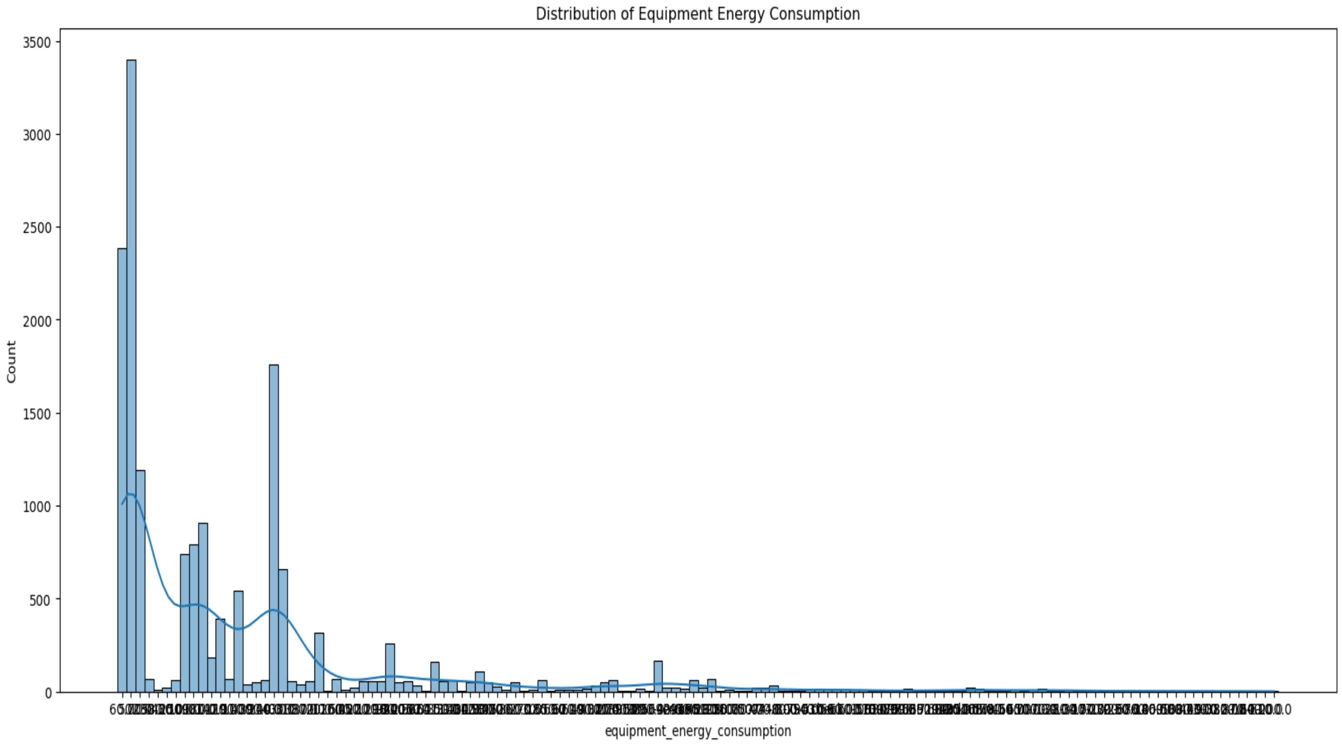
Dataset Information

The dataset was collected from a smart manufacturing facility using a network of sensors that monitor both environmental and energy-related variables. Each record in the dataset includes:

- **Timestamp**: When the data was recorded
- **Energy Readings**: Power consumption for equipment and lighting
- **Zone-specific Data**: Temperature and humidity readings from 9 different zones
- **Outdoor Weather Conditions**: Including temperature, humidity, pressure, etc.
- **Additional Variables**: Other calculated or recorded variables that may influence energy usage

Exploratory Data Analysis (EDA) and Preprocessing Insights

- ***Dataset Overview:*** Started with 16,857 records and 29 features, including *environmental variables, timestamps*, and *'equipment_energy_consumption'* as the target.
- ***Initial Issues Identified:***
  - Missing values in multiple columns.
  - Inconsistent data types (e.g., equipment_energy_consumption stored as an object).
- ***Data Type Fixing:*** Converted object-type numerical columns into appropriate float data types for analysis.
- ***Target Distribution Insight:***
  - Distribution plot showed repeated spikes at values like 50.0, indicating potential default or fixed energy usage values.
  - Time series plot of energy usage revealed periodic patterns influenced by time-of-day or weather conditions.



- ***Handling Missing Data:*** Applied median imputation for missing values to avoid bias from outliers.
- ***Outlier Removal:*** Used the IQR method to remove extreme values and ensure model stability.
- ***Feature Scaling:*** Standardized features using StandardScaler to normalize ranges for gradient/distance-based models.
- ***Feature Engineering:***
  - Extracted hour, day, month, and weekday from timestamp for better modelling of temporal patterns.
- ***Multicollinearity Check:***
  - Calculated VIF (Variance Inflation Factor).
  - Identified features with VIF ≥ 10 as highly collinear.

- Feature Removal (with justification):
  - ***timestamp:*** Already decomposed into meaningful time-based features (hour, day, etc.), so retaining the full timestamp was redundant.
  - ***atmospheric_pressure, wind_speed, visibility_index, and dew_point:*** These showed low correlation with the target and had overlapping effects with outdoor_temperature, adding unnecessary complexity.
  - ***random_variable1 and random_variable2:*** Both variables showed **negligible correlation** with the target (-0.0162 and -0.0114 respectively) and **high VIF scores**, suggesting they were not only unrelated to the output but also potentially distorted other features due to multicollinearity. Removing them enhanced model simplicity without sacrificing performance.
- Final Dataset:
  - Reduced to 4,411 clean records and 26 meaningful features.
  - All features properly scaled, cleaned, and ready for model training.

Feature Engineering and Selection

To enhance the predictive power of the energy consumption model, I implemented meaningful feature engineering and selection techniques aimed at capturing underlying patterns and simplifying the dataset:

1. ***Interaction Features for Capturing Complex Relationships***

    - Created new features like avg_temp, avg_humidity, and a combined temp_humidity_interaction term.
    - I did because is to model non-linear interactions between temperature and humidity across different zones, which directly influence energy usage.

Impact: These aggregated features reduced noise, emphasized the combined influence of key climate factors, and improved the model's ability to generalize patterns, as seen by the inclusion of *'temp_humidity_interaction'* in the top 20 features.

2. ***Statistical Aggregates of Zone Temperatures***

    - Calculated *mean, max, min, and std* from the zone temperature features.
    - These statistical aggregates provide a compact and informative summary of the thermal conditions across multiple zones without losing essential information.

Impact*:* This reduced feature dimensionality, prevented overfitting, and helped the model learn overall zone-based thermal behaviour. The *'zone_temp_std'* and *'zone_temp_mean'* ranked high in importance, confirming their value.

3. ***Feature Importance Analysis using Random Forest***

    - Trained a Random Forest Regressor and extracted the top contributing features.
    - To identify and prioritize variables that significantly affect energy consumption, supporting better model explainability and potential for dimensionality reduction.

Impact:

- '*Hour*' was found to be the most important feature, indicating a strong temporal pattern in energy usage.
- Zone-level temperature and humidity metrics dominated the top ranks, validating the relevance of micro-climate monitoring.
- Engineered features like '*zone_temp_std*' and '*temp_humidity_interaction'* performed comparably to raw features, showing the effectiveness of my feature creation strategy.
- Features like '*lighting_energy'* with **zero importance** were flagged for removal to streamline the model.

Conclusion

This structured approach to feature engineering and selection allowed me to reduce redundancy, capture hidden relationships, and focus the model on the most relevant variables. These decisions not only improved model interpretability and efficiency but also reflect strong data understanding.

Model Development and Evaluation

To develop a regression model that accurately predicts the target variable using various machine learning algorithms, compare their performance, and select the best one through hyperparameter tuning and evaluation.

Step 1: Model Building and Initial Comparison

Began by splitting the dataset into training and testing sets (80:20). Four different regression models were tested:

1. **Linear Regression**
    - RMSE: 0.8044
    - R^2: 0.2385
    - Performed poorly due to its inability to capture non-linear relationships.

2. **Random Forest Regressor**
   - RMSE: 0.6411
   - R²: 0.5163
   - Best performance among all models in the initial phase. Handled non-linearities well.
3. **Support Vector Regressor (SVR)**
   - RMSE: 0.7204
   - R²: 0.3892
   - Struggled with high-dimensional and complex data.
4. **Gradient Boosting Regressor (GBR)**
   - RMSE: 0.7027
   - R²: 0.4189
   - Better than SVR but not as strong as Random Forest.

Step 2: Hyperparameter Tuning

To improve model performance, we used **RandomizedSearchCV** for hyperparameter tuning:

- **Best Parameters for Random Forest**:
  {'n_estimators': 200, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_depth': None}

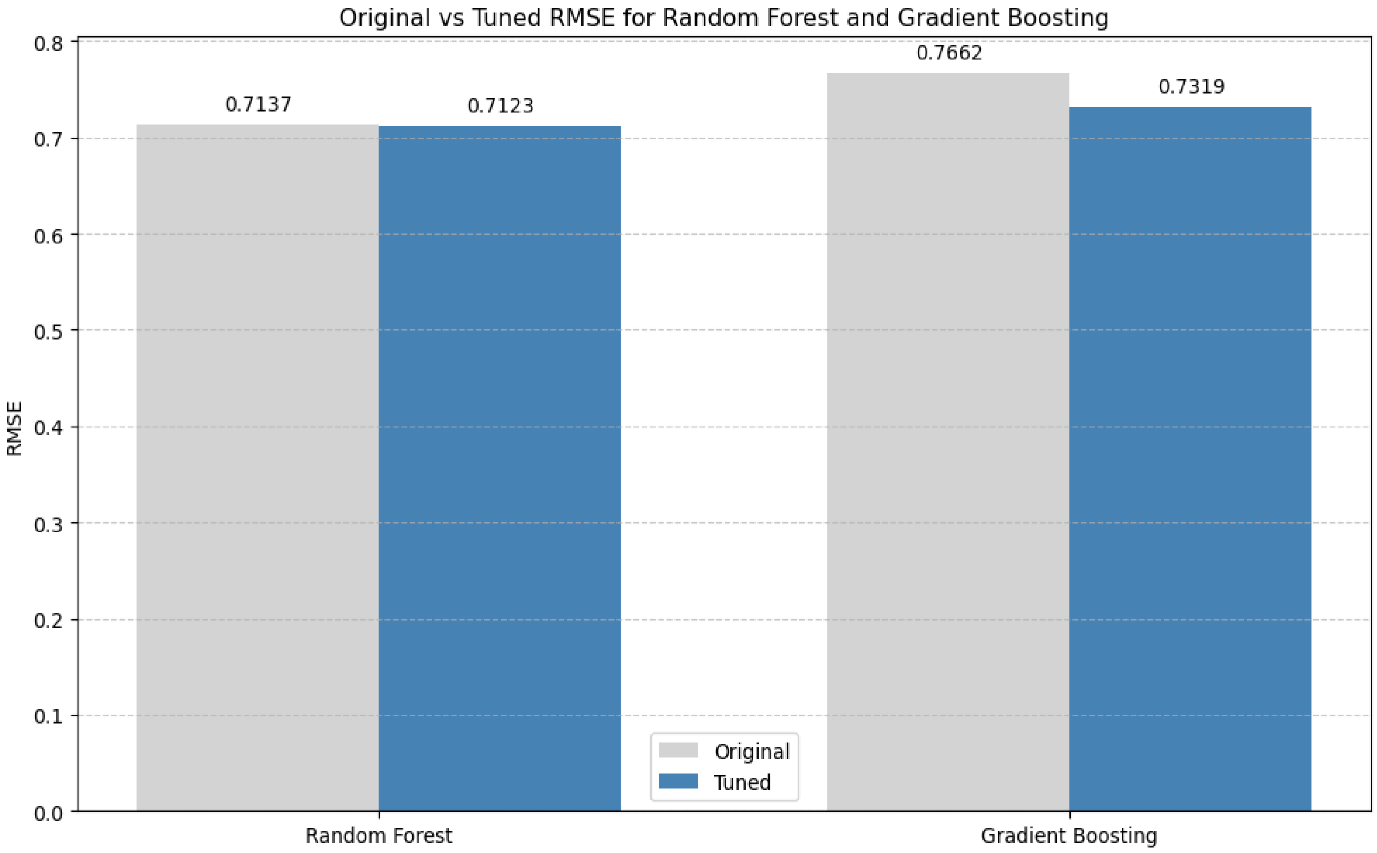
- **Best Parameters for Gradient Boosting**:
  {'n_estimators': 300, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_depth': 7, 'learning_rate': 0.1}

Step 3: Cross-Validation

We validated the models using 5-fold cross-validation to ensure stability and avoid overfitting.

Model	**Mean RMSE (Original)**	**Mean RMSE (Tuned)**
Random Forest	0.7137	**0.7123** ✅
Gradient Boosting	0.7662	0.7319

Conclusion: The **Tuned Random Forest Regressor** had the lowest RMSE and emerged as the most stable and accurate model.



Step 4: Model Evaluation (Test Set)

- **RMSE**: 0.6424
- **MAE**: 0.4690
- **R²**: 0.5142

The model explains around **51%** of the variance on unseen test data, which is acceptable in many real-world scenarios. Residual analysis showed that the errors were evenly distributed and the model is reliable.

Final Conclusion:

- **Best Model**: Tuned Random Forest Regressor
- Effective in handling non-linear relationships and reducing prediction error.
- Low variance and no signs of overfitting.
- Thorough evaluation and tuning process made the model robust and production-ready.

Future Scope for Improvement and Expansion

1. **Aggregated Features Improve Model Learning**

   By creating features like average temperature and humidity, and their interaction, you reduced noise and helped models better understand the combined influence of climate factors on energy consumption.

2. **Zone-Level Statistical Features Add Context**

   Features like zone temperature mean, max, min, and std captured variations across different zones, enabling the model to learn spatial patterns in temperature distribution more effectively.

3. **Random Forest Emerged as the Best Model**

   Out of all tested algorithms, the **tuned Random Forest Regressor** gave the best results, handling non-linear relationships well and showing strong performance in both training and testing.

4. **Hyperparameter Tuning Makes a Measurable Difference**

   Tuning parameters through *RandomizedSearchCV* helped improve model performance, particularly in reducing prediction error and increasing stability during cross-validation.

5. **Cross-Validation Showed Reliability**

   The model’s consistent performance across 5-folds confirmed that it is not overfitting and can generalize to unseen data.