Energy Consumption Forecasting

Overview

This project aims to predict energy consumption based on various features such as temperature, humidity, occupancy, and other environmental and operational factors. The prediction is done using machine learning models, and the best-performing model is tracked and logged with MLflow for experimentation and model management.

Objective

- Goal: To build an accurate forecasting model that predicts energy consumption.
- Primary Models:
  - Random Forest
  - XGBoost
  - HistGradientBoosting
- Evaluation Metrics:
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Percentage Error (MAPE)

The pipeline supports data preprocessing, feature engineering, model selection, hyperparameter optimization (using GridSearchCV), and logging all experiments with MLflow.

Key Files

1. Energy_consumption.csv: The dataset containing timestamped energy consumption data along with relevant features like temperature, humidity, occupancy, etc.
2. notebook.ipynb: The main Jupyter notebook that performs data processing, model training, evaluation, and MLflow logging.
3. requirements.txt: The list of dependencies required to run the project.

Setup Instructions

1. Dependencies

Install the necessary dependencies by running the following command:

pip install -r requirements.txt

2. MLflow Setup

Ensure MLflow is running for tracking experiments:
mlflow ui

- Access the MLflow interface at http://127.0.0.1:5000.

3. Dataset

The dataset used for this project is `Energy_consumption.csv`, which contains the following key columns:
- Timestamp: Date and time of the record.
- Temperature: The temperature recorded at that time.
- Humidity: The humidity recorded at that time.
- SquareFootage: The size of the building.
- Occupancy: Number of people occupying the building.
- EnergyConsumption: The target variable representing energy consumption.

4. Running the Project

To train models and evaluate their performance, follow these steps:

1. Load and preprocess the dataset using the preprocessing pipeline provided.
2. Use GridSearchCV to optimize the model hyperparameters.
3. Track the results of each model with MLflow, including metrics like RMSE and MAPE.
4. The best-performing model is saved in MLflow and can be retrieved for predictions on future data.

5. Key Scripts and Functions

a. Data Preprocessing and Feature Engineering

The data preprocessing pipeline includes scaling numerical features and encoding categorical features. Additionally, the pipeline calculates two derived features:

- Energy_per_sqft: Energy consumption per square foot of space.
- Energy_per_occupant: Energy consumption per person in the building.

b. Model Training

Multiple models (Random Forest, XGBoost, HistGradientBoosting) are trained and evaluated based on their performance metrics (RMSE, MAPE). Hyperparameter optimization is performed using GridSearchCV.

c. MLflow Logging

All models, parameters, and metrics are logged using MLflow. For each run:
- RMSE and MAPE are logged as key performance metrics.
- The best hyperparameters are stored and used to train the final models.

6. Model Evaluation

The GridSearchCV is used to identify the best model and hyperparameters. Performance is compared across models using cross-validation, and the results are logged in MLflow.

7. Predicting Energy Consumption

Once the best model is identified and logged, it can be used to predict future energy consumption. This is done using a separate inference pipeline where the best model is retrieved from MLflow, and predictions are made on new data.