

Project Name :Measure energy consumption of a building

Project author: MOHANRAJ S (810021106048)

Certainly, building a system to measure the energy consumption of a building using machine learning involves several key steps. Here's a simplified version:

- **Data Collection:**Gather data on the building's energy consumption. This data can come from utility bills, smart meters, or IoT sensors installed in the building.
- **Data Preprocessing:**Clean and preprocess the data, addressing missing values and outliers. Convert the data into a suitable format for machine learning.
- **Feature Engineering:**Identify and create features that can help the model understand energy consumption patterns. Consider factors like time of day, day of the week, temperature, and occupancy.
- **Selecting a Machine Learning Algorithm:**Choose a machine learning algorithm suitable for regression tasks. Linear regression is a good starting point.
- **Model Training:**Train the machine learning model using the preprocessed data.
- **Model Evaluation:**Assess the model's performance using metrics like Mean Absolute Error (MAE) or Mean Squared Error (MSE).
- **Analysis:**Analyze the results to understand how well the model is performing and whether any specific patterns or trends are identified.
- **Document Creation:**Create a document that details the process, including data sources, preprocessing, feature engineering, the chosen algorithm, training, evaluation results, and analysis findings.This document should serve as a record of the work performed during the project, making it easier to communicate the methodology and results to stakeholders or assessors.

Keep in mind that this is a simplified outline, and in a real-world scenario, the project might involve more complex data sources, advanced algorithms, and deeper analysis to optimize energy consumption measurement effectively.

Code:

Certainly, here's a simplified Python program to get you started on building a basic model for measuring energy consumption of a building using linear regression. Please note that this is a simplified example, and in practice, you'd require more data and sophisticated techniques for a reliable model.

```
import pandas as pd
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

Load your energy consumption data into a Pandas DataFrame

Replace 'your_data.csv' with your actual data file

```
data = pd.read_csv('your_data.csv')
```

Data preprocessing and feature engineering can be done here

Define your features (X) and target variable (y)

```
X = data[['Feature1', 'Feature2', ...]] # Add relevant features
```

```
y = data['EnergyConsumption'] # Assuming 'EnergyConsumption' is the target variable
```

Split the data into training and testing sets

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,  
random state=42)
```

Create and train a linear regression model

model = LinearRegression()

model.fit(X_train, y_train)

Make predictions on the test data

y_pred = model.predict(X_test)

Evaluate the model

mse = mean_squared_error(y_test, y_pred)

mae = mean_absolute_error(y_test, y_pred)

r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")

print(f"Mean Absolute Error: {mae}")

print(f"R-squared: {r2}")

You can now use this model for predictions and further analysis

Optionally, save the model for future use

import joblib

joblib.dump(model, 'energy_consumption_model.pkl') Remember to replace 'your_data.csv' with the path to your actual data file and add relevant features to X based on your dataset. You might also need to perform more extensive data preprocessing and feature engineering for a real-world application.