MEASURE ENERGY CONSUMPTION

312621243009 : Deepak.S

PHASE 3 - Document Submission

MODULES IMPORTED:

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean_squared_error, r2_score

import matplotlib.pyplot as plt

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
```

PANDAS:

PURPOSE: pandas is a powerful library for data manipulation and analysis. It provides data structures and functions for working with structured data, including CSV files, Excel spreadsheets, and SQL databases.

USAGE IN CODE: In your code, pandas is used for loading and preprocessing the dataset, as well as for data analysis and extraction.

sklearn.model_selection:

PURPOSE: The model_selection module from Scikit-Learn (or sklearn) provides tools for data splitting, cross-validation, and hyperparameter tuning. It's used for partitioning data into training and testing sets.

USAGE IN CODE: In your code, train_test_split is used to split the dataset into training and testing sets. This is a crucial step for training and evaluating machine learning models.

sklearn.ensemble:

PURPOSE: The ensemble module in Scikit-Learn contains ensemble methods for building and combining multiple machine learning models, such as Random Forests and Gradient Boosting.

USAGE IN CODE: In your code, you've imported RandomForestRegressor, which is a specific ensemble model used for regression tasks. It's employed for building a predictive model to estimate energy consumption.

sklearn.metrics:

PURPOSE: The metrics module in Scikit-Learn provides various metrics for evaluating machine learning models. Common metrics include mean squared error (MSE), R-squared (R2), accuracy, precision, and recall.

USAGE IN CODE: In your code, mean_squared_error and r2_score are used to evaluate the performance of your regression model. mean_squared_error calculates the mean squared error between predicted and actual values, while r2_score computes the R-squared coefficient, which measures the goodness of fit of your model.

matplotlib.pyplot:

PURPOSE: The matplotlib library is widely used for creating data visualizations and plots. pyplot is a subpackage of matplotlib that provides a MATLAB-style interface for creating static, animated, or interactive visualizations in Python.

USAGE IN CODE: In your code, matplotlib.pyplot is used to create a scatter plot to visualize the relationship between actual and predicted energy consumption values. This helps in assessing the model's predictive performance.

These modules are essential tools for working with data, building machine learning models, and evaluating their performance. They are widely used in data science and machine learning projects to perform tasks like data preprocessing, model training, and result visualization.

PRE PROCESSING:

We have wrote the code by importing the module above and using the data set which has been provided in the capital and the code has been optimized and able to run and give the output for the required operation and it is possible to make the user requirements satisfied.

```
# Step 1: Load the Dataset
file_path = r'C:\Users\Deepak Dev\Downloads\hourly_energy_consumption.csv'
data = pd.read_csv(file_path)

# Step 2: Preprocess the Data (e.g., handle missing values)
data.fillna(data.mean(), inplace=True)

# Step 3: Define Features and Target
X = data[['feature1', 'feature2', '...']]  # Replace 'feature1', 'feature2', '...' with actual features
y = data['energy_consumption']
```

PROCESSING:

In the above code the data set has been processed and able to give the required output as per user input.

```
# Step 4: 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)

# Step 5: Create and Train a Random Forest Regression Model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Step 6: Make Predictions
y_pred = model.predict(X_test)

# Step 7: Evaluate the Model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print(f"R-squared (R2) Score: {r2}")
```

Now,The code is able to work on the data set by using the neural network above.

THE END PROCESSING:

Now the required output is been obtained.

```
print(f"Mean Squared Error: {mse}")
print(f"R-squared (R2) Score: {r2}")

# Step 8: Visualize Predictions (Optional)
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Energy Consumption")
plt.ylabel("Predicted Energy Consumption")
plt.title("Actual vs. Predicted Energy Consumption")
plt.show()

```

Hence,The output for the above code is

```
Mean Squared Error: [MSE_VALUE]
R-squared (R2) Score: [R2_VALUE]
```