In [1]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
import numpy as np

# Load the energy consumption data
```

In [2]:

```
# Split the data into input features and target variable
X = data.drop('consumption', axis=1)
```

In [3]:

```
# 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, rando
```

In [4]:

```
# Train a Random Forest regression model
model = RandomForestRegressor()
model.fit(X_train, y_train)
```

Out[4]:

```
▾ RandomForestRegressor

RandomForestRegressor()
```

In [5]:

```
# Make predictions on the test set
y_pred = model.predict(X_test)
```

In [6]:

```
# Calculate the root mean squared error
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
print(f"Root Mean Squared Error: {rmse}")
```

```
Root Mean Squared Error: 13.037690804326402
```

In [7]:

```
# Perform energy optimization using the trained model
def optimize_energy(model, input_data, threshold):
    predicted_consumption = model.predict(input_data)
    optimized_consumption = np.where(predicted_consumption > threshold, predic
    return optimized_consumption
```

In [8]:

```
input_data = pd.DataFrame([[40, 35, 1]], columns=['temperature', 'humidity', '
threshold = 30  # Adjust the threshold value based on your optimization criter
```

In [9]:

```
Optimized energy consumption: [93.67969308]
```

In [ ]:

In [ ]: