SOURCE CODE:

```
# Energy Efficiency Prediction using
Random Forest
import pandas as pd
import numpy as np
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

# Load dataset
# You can download the UCI Energy
Efficiency dataset from:
#
https://archive.ics.uci.edu/ml/datas
ets/Energy+efficiency
data =
pd.read_excel("ENB2012_data.xlsx")

# Rename columns for simplicity
data.columns =
['RelativeCompactness',
'SurfaceArea', 'WallArea',
'RoofArea',
                'OverallHeight',
'Orientation', 'GlazingArea',
'GlazingAreaDistribution',
                'HeatingLoad',
'CoolingLoad']
```

```
# Features and targets
X = data.iloc[:, 0:8]  # Input
features
y = data['HeatingLoad']  # Target:
Heating Load

# Train-test split
X_train, X_test, y_train, y_test =
train_test_split(X, y,
test_size=0.2, random_state=42)

# Model training
model =
RandomForestRegressor(n_estimators=1
00, random_state=42)
model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)

# Evaluation
mse = mean_squared_error(y_test,
y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error:
{mse:.2f}")
print(f"R2 Score: {r2:.2f}")
```

```
# Plotting actual vs predicted
plt.figure(figsize=(8,5))
plt.scatter(y_test, y_pred,
color='green')
plt.plot([y_test.min(),
y_test.max()], [y_test.min(),
y_test.max()], 'r--')
plt.xlabel("Actual Heating Load")
plt.ylabel("Predicted Heating Load")
plt.title("Actual vs Predicted
Heating Load")
plt.grid(True)
plt.show()
```

Output: