

```
Energy2.py X
Energy2.py > ...
1 import pandas as pd
2 from sklearn.model_selection import train_test_split, GridSearchCV
3 from sklearn.preprocessing import StandardScaler
4 from sklearn.ensemble import RandomForestRegressor
5 from sklearn.tree import DecisionTreeRegressor
6 from sklearn.metrics import mean_squared_error, r2_score
7
8 # Load the dataset
9 file_path = 'C:/Users/2273581/Downloads/Energy_dataset.csv'
10 df = pd.read_csv(file_path)
11
12 # Define the features (X) and target (y)
13 X = df[['energy_consumption_kwh', 'peak_hours_usage', 'off_peak_usage', 'renewable_energy_pct', 'household_size', 'temperature_c']]
14 y = df['billing_amount']
15
16 # Split the dataset into training and testing sets
17 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
18
19 # Standardize the features
20 scaler = StandardScaler()
21 X_train = scaler.fit_transform(X_train)
22 X_test = scaler.transform(X_test)
23
24 # Create a Decision Tree Regressor
25 dt_regressor = DecisionTreeRegressor()
26
27 # Create a GridSearchCV object
28 grid_search = GridSearchCV(dt_regressor, {'max_depth': [10, 20, 30], 'min_samples_leaf': [2, 4, 8], 'min_samples_split': [10, 20, 30]}, cv=5)
29
30 # Fit the model
31 grid_search.fit(X_train, y_train)
32
33 # Print the best parameters
34 print('Best parameters for Decision Tree: ', grid_search.best_params_)
35
36 # Print the best cross-validation score
37 print('Best cross-validation score for Decision Tree: ', grid_search.best_score_)
38
39 # Print the Mean Squared Error
40 print('Decision Tree - Mean Squared Error: ', mean_squared_error(X_test, y_test))
41
42 # Print the R-squared score
43 print('Decision Tree - R-squared: ', r2_score(X_test, y_test))
```

Best parameters for Decision Tree: {'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 10}
Best parameters for Decision Tree: {'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 10}
Best cross-validation score for Decision Tree: -0.075061928680156
Decision Tree - Mean Squared Error: 17702.50311737769
Decision Tree - Mean Squared Error: 17702.50311737769
Decision Tree - R-squared: -0.02891041950058204

```
Energy2.py X
Energy2.py > ...
1 import pandas as pd
2 from sklearn.model_selection import train_test_split, GridSearchCV
3 from sklearn.preprocessing import StandardScaler
4 from sklearn.ensemble import RandomForestRegressor
5 from sklearn.tree import DecisionTreeRegressor
6 from sklearn.metrics import mean_squared_error, r2_score
7
8 # Load the dataset
9 file_path = 'C:/Users/2273581/Downloads/Energy_dataset.csv'
10 df = pd.read_csv(file_path)
11
12 # Define the features (X) and target (y)
13 X = df[['energy_consumption_kwh', 'peak_hours_usage', 'off_peak_usage', 'renewable_energy_pct', 'household_size', 'temperature_c']]
14 y = df['billing_amount']
15
16 # Split the dataset into training and testing sets
17 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
18
19 # Standardize the features
20 scaler = StandardScaler()
21 X_train = scaler.fit_transform(X_train)
22 X_test = scaler.transform(X_test)
23
24 # Create a Random Forest Regressor
25 rf_regressor = RandomForestRegressor()
26
27 # Create a GridSearchCV object
28 grid_search = GridSearchCV(rf_regressor, {'max_depth': [10, 20, 30], 'min_samples_leaf': [2, 4, 8], 'min_samples_split': [10, 20, 30], 'n_estimators': [100, 200, 300]}, cv=5)
29
30 # Fit the model
31 grid_search.fit(X_train, y_train)
32
33 # Print the best parameters
34 print('Best parameters for Random Forest: ', grid_search.best_params_)
35
36 # Print the best cross-validation score
37 print('Best cross-validation score for Random Forest: ', grid_search.best_score_)
38
39 # Print the Mean Squared Error
40 print('Random Forest - Mean Squared Error: ', mean_squared_error(X_test, y_test))
41
42 # Print the R-squared score
43 print('Random Forest - R-squared: ', r2_score(X_test, y_test))
```

[CV] END max_depth=30, min_samples_leaf=4, min_samples_split=10, n_estimators=200; total time= 27.4s
[CV] END max_depth=30, min_samples_leaf=4, min_samples_split=10, n_estimators=200; total time= 27.3s
Best parameters for Random Forest: {'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 200}
Best cross-validation score for Random Forest: -0.006132373396719126
Random Forest - Mean Squared Error: 17343.91378899142
Random Forest - R-squared: -0.008068378471019155
Fitting 3 folds for each of 36 candidates, totalling 108 fits