[42]
✓ 0s

```
# Predict Water Footprint
y_pred = rf.predict(X_scaled)

df1["Predicted_Footprint"] = y_pred
print(df1[["Household", "Water_Footprint", "Predicted_Footprint"]].head())
```

```
   Household  Water_Footprint  Predicted_Footprint
0  -1.566699        -1.156140            -0.903986
1  -0.870388         1.286256             0.145683
2  -0.522233        -0.322257            -0.311606
3  -0.174078         0.989654             0.453313
4   0.174078        -0.371453            -0.062368
```

[43]
✓ 0s

```
top_households = df1.nlargest(3, "Predicted_Footprint")[["Household", "Predicted_Footprint"]]
print("High Footprint Households:")
print(top_households)
```

```
High Footprint Households:
   Household  Predicted_Footprint
7   1.218544             1.151821
3  -0.174078             0.453313
9  -1.218544             0.402762
```

[44]
✓ 0s

```
activity_cols = ["Bathing(L)", "Cooking(L)", "Washing(L)", "Gardening(L)", "Drinking(L)"]

df1["Max_Activity"] = df1[activity_cols].idxmax(axis=1)
print(df1[["Household", "Max_Activity"]].head())
```

```
   Household  Max_Activity
0  -1.566699   Drinking(L)
1  -0.870388  Gardening(L)
2  -0.522233    Cooking(L)
3  -0.174078    Bathing(L)
4   0.174078    Bathing(L)
```

[45]
✓ 0s

```python
def recommend(row):
    if row["Max_Activity"] == "Bathing(L)":
        return "Reduce shower time or use low-flow showerheads."
    elif row["Max_Activity"] == "Cooking(L)":
        return "Use water-efficient cooking methods."
    elif row["Max_Activity"] == "Washing(L)":
        return "Run washing machine with full loads only."
    elif row["Max_Activity"] == "Gardening(L)":
        return "Adopt drip irrigation / reuse household water for gardening."
    elif row["Max_Activity"] == "Drinking(L)":
        return "Check for leaks / encourage mindful drinking."
    else:
        return "Maintain current usage."

df1["Recommendation"] = df1.apply(recommend, axis=1)
print(df1[["Household", "Max_Activity", "Recommendation"]].head())
```

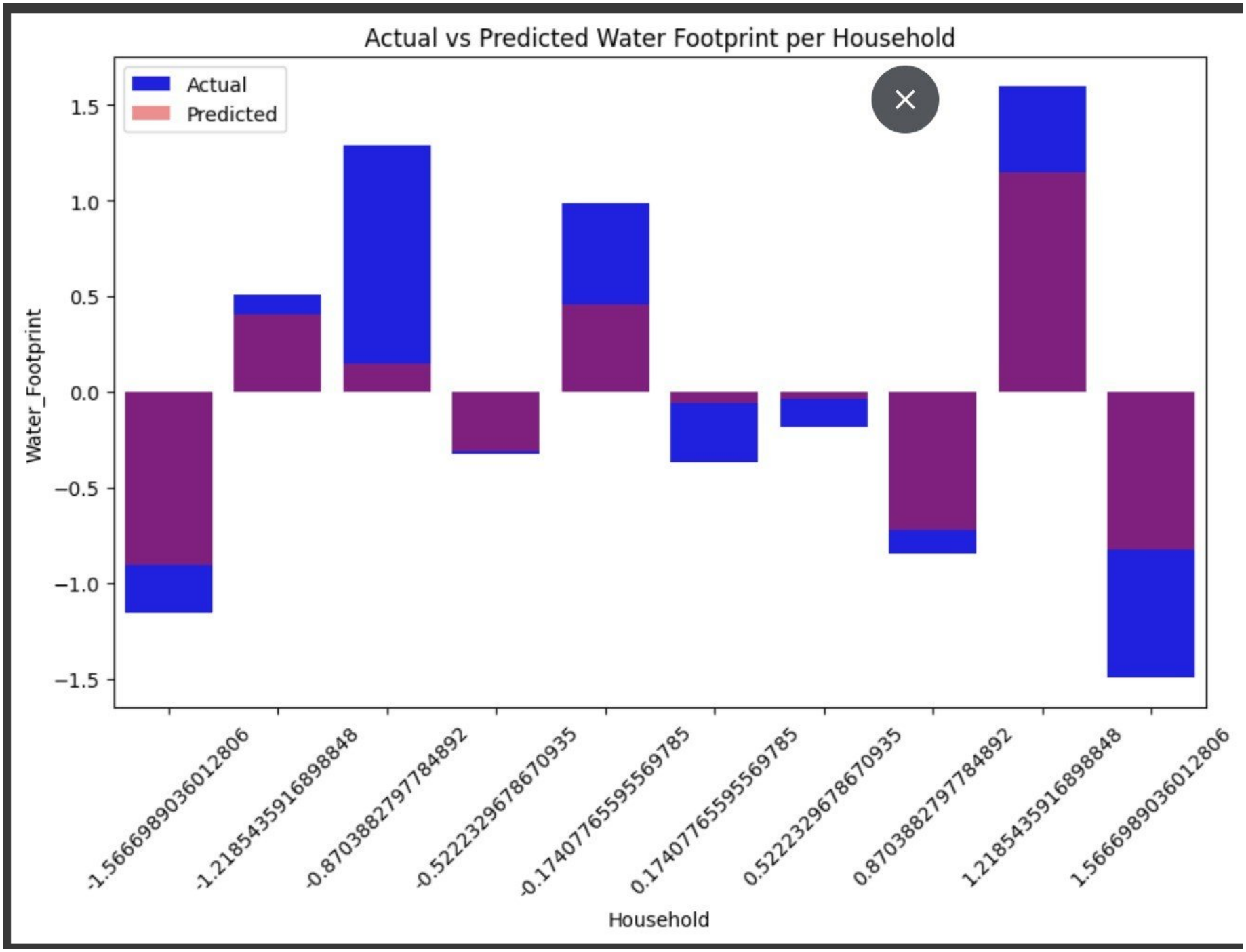
0s

	Household	Max_Activity	Recommendation
0	-1.566699	Drinking(L)	Check for leaks / encourage mindful drinking.
1	-0.870388	Gardening(L)	Adopt drip irrigation / reuse household water ...
2	-0.522233	Cooking(L)	Use water-efficient cooking methods.
3	-0.174078	Bathing(L)	Reduce shower time or use low-flow showerheads.
4	0.174078	Bathing(L)	Reduce shower time or use low-flow showerheads.

[46]
0s

```
import matplotlib.pyplot as plt
import seaborn as sns

# Actual vs Predicted
plt.figure(figsize=(10,6))
sns.barplot(x="Household", y="Water_Footprint", data=df1, color="blue", label="Actual")
sns.barplot(x="Household", y="Predicted_Footprint", data=df1, color="red", alpha=0.5, label="Predicted")
plt.xticks(rotation=45)
plt.title("Actual vs Predicted Water Footprint per Household")
plt.legend()
plt.show()
```


Actual vs Predicted Water Footprint per Household
Actual
Predicted
Water_Footprint
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
-1.5666989036012806
-1.2185435916898848
-0.8703882797784892
-0.5222329678670935
-0.1740776559569785
0.1740776559569785
0.5222329678670935
0.8703882797784892
1.2185435916898848
1.5666989036012806
Household