In [1]:
```
pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\mrmua\new folder\lib\site-pac
kages (2.0.3)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: scipy in c:\users\mrmua\new folder\lib\site-packa
ges (from xgboost) (1.9.1)
Requirement already satisfied: numpy in c:\users\mrmua\new folder\lib\site-packa
ges (from xgboost) (1.24.4)
```

In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import r2_score
```

In [3]:
```
file_path = r"D:\CV things\ML projects\Train.csv"  # Specify the file path
dt = pd.read_csv(file_path)  # Read the CSV file into a DataFrame

# Use 'df' for further processing
print(dt.head())  # Displaying the first few rows as an example
```

```
  Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0           FDA15         9.30          Low Fat         0.016047
1           DRC01         5.92          Regular         0.019278
2           FDN15        17.50          Low Fat         0.016760
3           FDX07        19.20          Regular         0.000000
4           NCD19         8.93          Low Fat         0.000000

              Item_Type  Item_MRP Outlet_Identifier  \
0                 Dairy  249.8092            OUT049
1           Soft Drinks   48.2692            OUT018
2                  Meat  141.6180            OUT049
3  Fruits and Vegetables  182.0950            OUT010
4             Household   53.8614            OUT013

   Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
0                       1999      Medium               Tier 1
1                       2009      Medium               Tier 3
2                       1999      Medium               Tier 1
3                       1998         NaN               Tier 3
4                       1987        High               Tier 3

        Outlet_Type  Item_Outlet_Sales
0  Supermarket Type1          3735.1380
1  Supermarket Type2           443.4228
2  Supermarket Type1          2097.2700
3      Grocery Store           732.3800
4  Supermarket Type1           994.7052
```

In [4]:
```
# Handle missing values
dt['Item_Weight'].fillna(dt['Item_Weight'].mean(), inplace=True)
dt['Outlet_Size'].fillna(dt['Outlet_Size'].mode()[0], inplace=True)
```

In [5]:
```
# Feature Engineering
dt['Outlet_Years'] = 2022 - dt['Outlet_Establishment_Year']
dt['New_Item_Type'] = dt['Item_Identifier'].apply(lambda x: x[:2])
dt['New_Item_Type'] = dt['New_Item_Type'].map({'FD':'Food', 'NC':'Non-Consumable
```

In [6]:
```
# Define categorical columns for encoding
cat_cols = ['Item_Fat_Content', 'Item_Type', 'Outlet_Size', 'Outlet_Location_Typ
```

In [7]:
```
# Encode categorical columns
dt_encoded = pd.get_dummies(dt, columns=cat_cols)
```

In [8]:
```
# Define features and target variable
X = dt_encoded.drop(columns=['Outlet_Establishment_Year', 'Item_Identifier', 'Ou
y = np.log1p(dt_encoded['Item_Outlet_Sales'])  # Log transformation of target va
```

In [9]:
```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_
```

In [10]:
```
# RandomForestRegressor
rf = RandomForestRegressor()
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [5, 10, 15, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

rf_random = RandomizedSearchCV(estimator=rf, param_distributions=param_grid, n_i
rf_random.fit(X_train, y_train)

best_rf = rf_random.best_estimator_
y_pred_rf = best_rf.predict(X_test)
r2_rf = r2_score(y_test, y_pred_rf)
print(f"R2 Score (Random Forest): {r2_rf}")
```

```
R2 Score (Random Forest): 0.7256058769473517
```