```
# Import necessary libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Generate some sample data
np.random.seed(42)
square_feet = np.random.randint(1000, 3000, 100)
num_bedrooms = np.random.randint(1, 5, 100)
housing_prices = 100000 + 200 * square_feet + 50000 * num_bedrooms + np.random.normal(0, 10000, 100)

# Create a DataFrame
data = pd.DataFrame({'SquareFeet': square_feet, 'Bedrooms': num_bedrooms, 'Price': housing_prices})

# Split the data into features (X) and target variable (y)
X = data[['SquareFeet', 'Bedrooms']]
y = data['Price']

# 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, random_state=42)

# Initialize the Gradient Boosting Regressor
model = GradientBoostingRegressor()

# Fit the model to the training data
model.fit(X_train, y_train)

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

# Evaluate the model
mse = mean_squared_error(y_test, predictions)
print(f'Mean Squared Error: {mse}')

# Plotting actual vs predicted prices
plt.scatter(y_test, predictions)
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual Prices vs Predicted Prices")
plt.show()
```