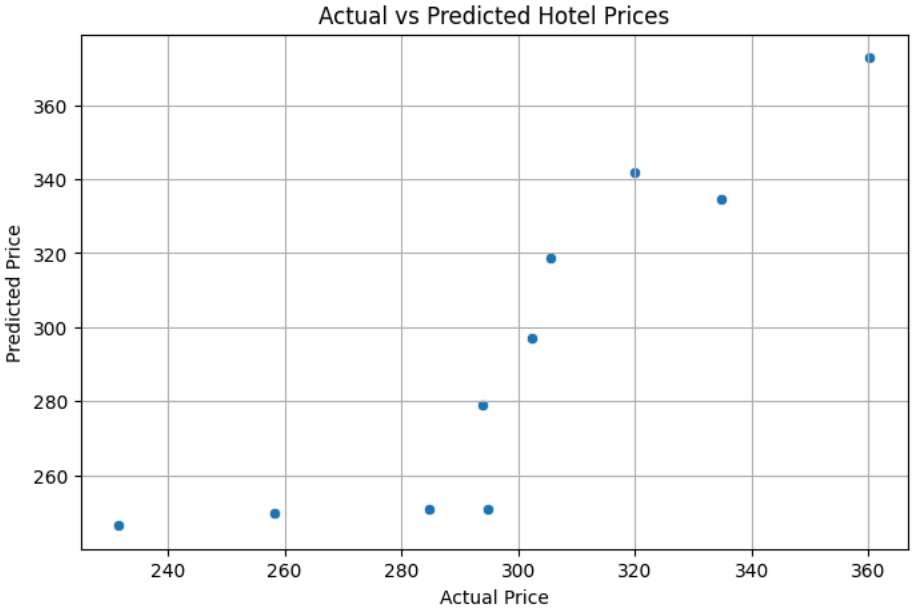
🔷 2. Load the Dataset

✅ X holds the input data; y holds the class we want to predict.

```
# 1. Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error
import matplotlib.pyplot as plt
import seaborn as sns

# 2. Read the dataset
df = pd.read_csv("london_hotels_structured.csv")

# 3. Observe the first few rows
print(df.head())

# 4. Define target and feature variables
# Target variable: Price_Per_Night
# Input features: Rating, Distance_from_City_Center, Review_Score
X = df[["star_rating", "distance_to_city_center", "review_score"]]
y = df["price"]

# 5. Split the data into training and test sets (80% training, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 6. Create and train the Decision Tree model
model = DecisionTreeRegressor(random_state=42)
model.fit(X_train, y_train)

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

# 8. Compare predictions with actual values (using MAE)
mae = mean_absolute_error(y_test, y_pred)
print(f"Mean Absolute Error: {mae:.2f} GBP")

# 9. Plot comparing actual vs predicted values
plt.figure(figsize=(8, 5))
sns.scatterplot(x=y_test, y=y_pred)
plt.xlabel("Actual Price")
plt.ylabel("Predicted Price")
plt.title("Actual vs Predicted Hotel Prices")
plt.grid(True)
plt.show()
```

```
   star_rating  review_score  distance_to_city_center  is_near_tube  \
0            5           7.9                     2.74             0
1            3           8.0                     2.93             0
2            5           7.6                     0.88             1
3            5           7.6                     7.25             0
4            3           8.5                     1.55             0

   breakfast_included  weekend   price
0                   0        0  334.47
1                   1        0  249.93
2                   1        0  372.78
3                   0        0  282.88
4                   1        0  272.48
Mean Absolute Error: 16.88 GBP
```




```
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split

# 1. Read the data
df = pd.read_csv("london_hotels_structured.csv")

# 2. Separate features and target variable
X = df.drop("price", axis=1)
y = df["price"]

# 3. Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 4. Create and train the model
model = DecisionTreeRegressor()
model.fit(X_train, y_train)

# 5. Get hotel details from the user (manual input)
print("Please enter the hotel features:")
star_rating = int(input("Star Rating (1-5): "))
review_score = float(input("Review Score (0.0 - 10.0): "))
distance = float(input("Distance to City Center (km): "))
is_near_tube = int(input("Is Near Tube? (1: Yes, 0: No): "))
breakfast_included = int(input("Breakfast Included? (1: Yes, 0: No): "))
weekend = int(input("Is it a Weekend? (1: Yes, 0: No): "))

# 6. Pass the input to the model for prediction
input_data = pd.DataFrame([[star_rating, review_score, distance, is_near_tube, breakfast_included, weekend]],
                          columns=X.columns)

predicted_price = model.predict(input_data)

# 7. Show the result
print(f"\n💰 Estimated Hotel Price: £{predicted_price[0]:.2f}")
```

```
Please enter the hotel features:
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
/tmp/ipython-input-3-1355523881.py in <cell line: 0>()
     19 # 5. Get hotel details from the user (manual input)
     20 print("Please enter the hotel features:")
---> 21 star_rating = int(input("Star Rating (1-5): "))
     22 review_score = float(input("Review Score (0.0 - 10.0): "))
     23 distance = float(input("Distance to City Center (km): "))

 1 frames
/usr/local/lib/python3.11/dist-packages/ipykernel/kernelbase.py in _input_request(self, prompt, ident, parent,
```