Mini Project: London Hotel Price Prediction

A linear regression project to predict hotel prices across London based on key features

Y by Yonca Kurt

📘 Mini Project Code Steps

- [ ] 📂 Load the Dataset
- [ ] 🧹 Check and Clean Missing Values
- [ ] 📊 Select Features (Independent Variables)
- [ ] 🧠 Build and Train Your Linear Regression Model
- [ ] 🔮 Make Predictions
- [ ] 📈 Visualize Predictions vs Actual Values
- [ ] 📌 Predict the Price of a New Hotel

Project Objective

Build a Linear Regression model to predict the price of a hotel room per night in London based on various features.

- Location
- Distance to tourist attractions
- Crime rate
- Star rating
- Number of reviews
- Guest ratings
- Room features

✅ Step 1: Load the Dataset

Start by loading the housing dataset using Pandas. This dataset contains real features like area size, number of bedrooms, and price.

```
import pandas as pd

df = pd.read_csv('/content/london_hotels.csv') # adjust the path if needed
df.head()
```

Download House Prices Dataset (CSV)

✅ Step 1: Load the Dataset (from Google Drive)

Start by loading the housing dataset using Pandas. This dataset contains real features like area size, number of bedrooms, and price.

```
# Connect Google Drive to access files
from google.colab import drive
drive.mount('/content/drive')

import pandas as pd

# Set the path to the CSV file
file_path = '/content/drive/My Drive/ai-camp/london_hotels.csv'

# Readt the CSV file
df = pd.read_csv(file_path)

# Display the first 5 rows of the dataset
df.head()
```

Download House Prices Dataset (CSV)

🧹 Step 2: Explore & Clean the Data

Real-world data is often messy. We'll check for missing values or outliers and clean the dataset to ensure accurate predictions.

Check for missing values, column types, and rename the columns if needed.

```
# Check the shape of the dataset (rows, columns)
print("Shape of the dataset:", df.shape)

df.info() # Gives Structural Information About the Dataset
df.describe() # Gives Statistical Summary
df.isnull().sum()
```

📊 Step 3: Select Features (Independent Variables)

Decide which features (X) you want to use to predict the target variable (y = price).

```
X = df[['star_rating', 'review_score', 'distance_to_city_center','is_near_tube','breakfast_included','weekend']]
y = df['price']
```

✅ Step 4: Split the Data

Split into training and testing sets.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

🧠 Step 5: Build and Train Your Model

Use Linear Regression to train your model.

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)

print("Coefficients:", model.coef_)
print("Intercept:", model.intercept_)
```

🔮 Step 6: Make Predictions

Use the trained model to predict the target values for test data

```
# Make Predictions

y_pred = model.predict(X_test)
```

🔮 Step 7: Evaluate the Model

Now it's time to test your model! Compare predictions with actual values using common regression metrics

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("Mean Absolute Error:", mae)
print("R2 Score:", r2)
```

✅ Step 7: Let's Evaluate the Model

- What was the R^2 value of your model?
- Is it better or worse than you expected?
- Why do you think it turned out that way?
- How might the weekend effect influence the price?
- Estimate the 3 most important things that influence the price!

ⓘ Conclusion

Good data leads to good models. Now it's your turn! Can you experiment with different features to further improve this score?

📈 Step 8: Visualize Results

Create plots to compare predicted vs actual prices. Visualization helps you understand how well your model is performing.

```
import matplotlib.pyplot as plt

plt.scatter(y_test, y_pred, color='blue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted Hotel Prices')
plt.grid(True)
plt.show()
```

📌 Step 9: *Try New Data!* Predict a New House Price

Enter the features of a new hotel and let your model predict the price.
This is your chance to test the model with data that wasn't in the original dataset.
See if the result feels realistic, and reflect on how well the model generalizes!

```
new_house = pd.DataFrame({
    'house_age': [15],
    'mrt_distance': [200],
    'convenience_stores': [4]
})
predicted_price = model.predict(new_house)
print(f"Predicted Price for the new house: ${predicted_price[0]:.2f}")
```

```
new_house = pd.DataFrame({
    'house_age': [10],
    'mrt_distance': [300],
    'convenience_stores': [5]
})
predicted_price = model.predict(new_house)
print(f"Predicted Price for the new house: ${predicted_price[0]:.2f}")
```