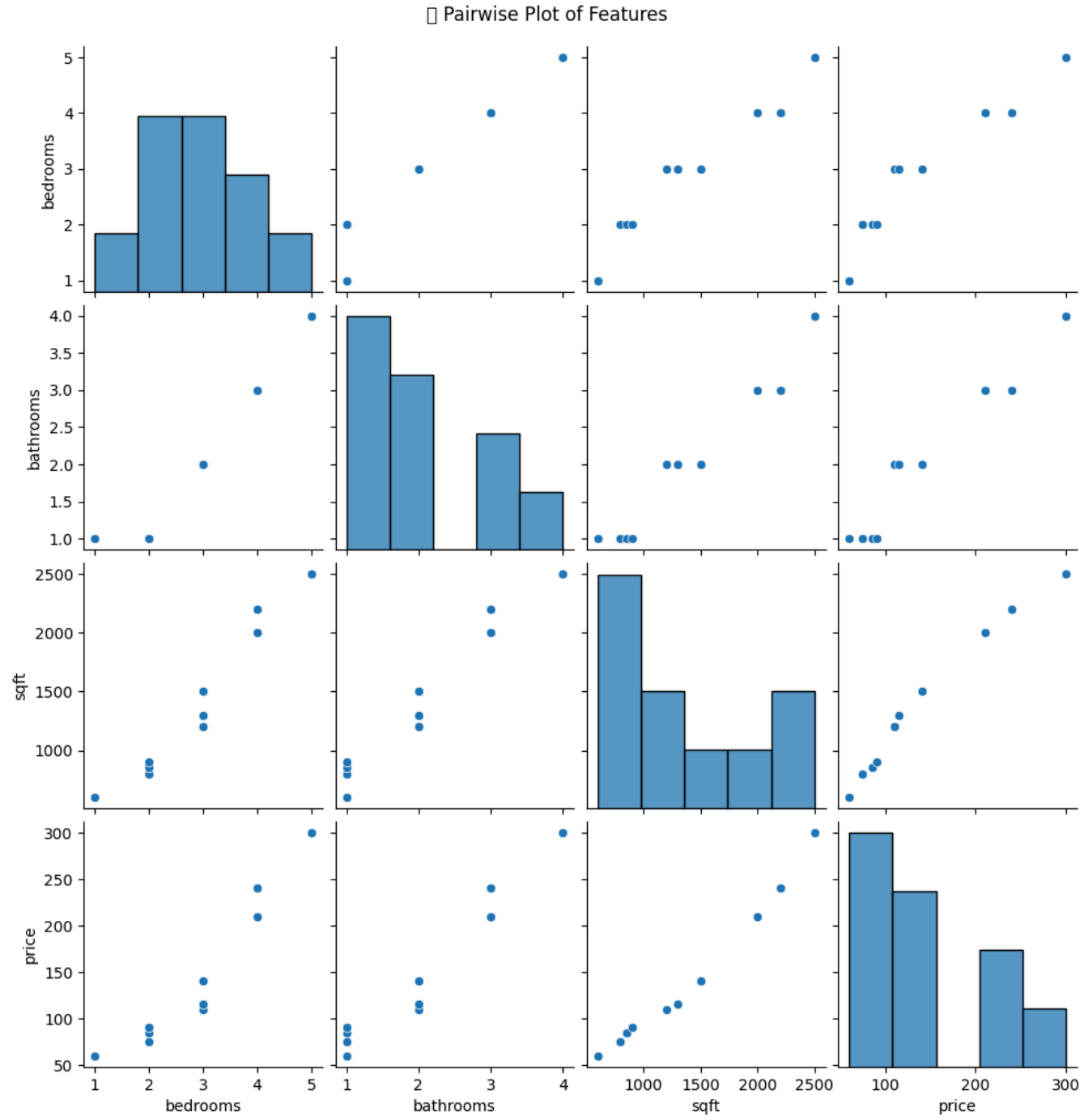
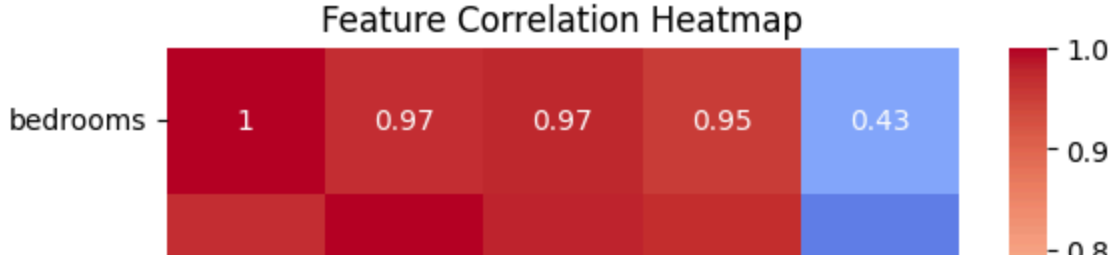
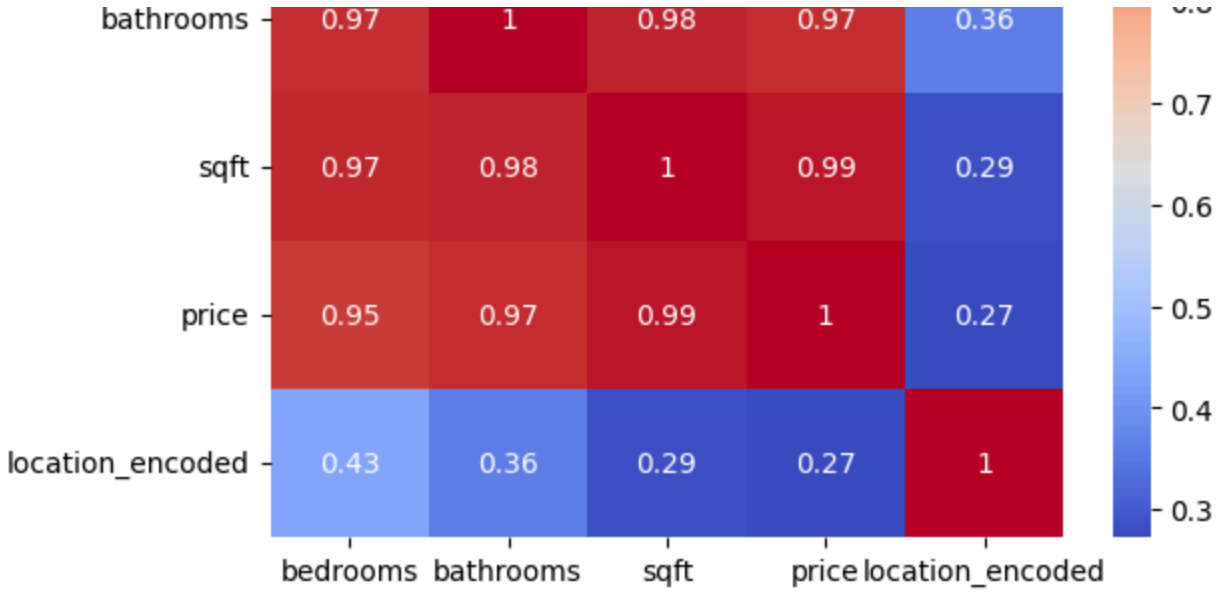
```
# ------------------------------------------
# 1. Install Required Libraries
# ------------------------------------------
!pip install -q scikit-learn pandas matplotlib seaborn gradio

# ------------------------------------------
# 2. Import Libraries
# ------------------------------------------
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import gradio as gr

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder

# ------------------------------------------
# 3. Create Simple Dataset
# ------------------------------------------
data = {
    'bedrooms': [2, 3, 4, 3, 5, 4, 2, 1, 3, 2],
    'bathrooms': [1, 2, 3, 2, 4, 3, 1, 1, 2, 1],
    'sqft': [800, 1200, 2000, 1500, 2500, 2200, 850, 600, 1300, 900],
    'location': ['urban', 'urban', 'suburban', 'rural', 'urban', 'suburban', 'rural', 'rura
    'price': [75, 110, 210, 140, 300, 240, 85, 60, 115, 90]  # in ₹ lakhs
}

df = pd.DataFrame(data)

# ------------------------------------------
# 4. Encode Categorical Column
# ------------------------------------------
le = LabelEncoder()
df['location_encoded'] = le.fit_transform(df['location'])  # e.g., urban -> 2, rural -> 1, 

# ------------------------------------------
# 5. Define Features and Target
# ------------------------------------------
X = df[['bedrooms', 'bathrooms', 'sqft', 'location_encoded']]
y = df['price']

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# ------------------------------------------
# 6. Visualization (Optional)
# ------------------------------------------
sns.pairplot(df[['bedrooms', 'bathrooms', 'sqft', 'price']])
plt.suptitle("📊 Pairwise Plot of Features", y=1.02)
plt.show()

sns.heatmap(df.select_dtypes(include='number').corr(), annot=True, cmap="coolwarm")
plt.title("Feature Correlation Heatmap")
plt.show()

# ------------------------------------------
```

```
# 7. Train Regression Models
# ------------------------------------------
lr = LinearRegression()
lr.fit(X_train, y_train)

ridge = Ridge(alpha=1.0)
ridge.fit(X_train, y_train)

lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)

# ------------------------------------------
# 8. Evaluate Models
# ------------------------------------------
models = {'Linear': lr, 'Ridge': ridge, 'Lasso': lasso}

for name, model in models.items():
    y_pred = model.predict(X_test)
    print(f"📊 {name} Regression Results:")
    print("  MSE:", round(mean_squared_error(y_test, y_pred), 2))
    print("  R² Score:", round(r2_score(y_test, y_pred), 2))
    print()

# ------------------------------------------
# 9. Gradio Deployment (using Ridge model)
# ------------------------------------------
def predict_price(bedrooms, bathrooms, sqft, location):
    try:
        loc_encoded = le.transform([location])[0]
    except:
        return "❌ Unknown location. Please select from the dropdown."

    input_data = [[bedrooms, bathrooms, sqft, loc_encoded]]
    prediction = ridge.predict(input_data)[0]
    return f"🏷️ Estimated House Price: ₹{round(prediction, 2)} lakhs"

# Interface
interface = gr.Interface(
    fn=predict_price,
    inputs=[
        gr.Slider(1, 5, step=1, label="Number of Bedrooms"),
        gr.Slider(1, 4, step=1, label="Number of Bathrooms"),
        gr.Slider(500, 3000, step=50, label="Square Footage"),
        gr.Dropdown(choices=le.classes_.tolist(), label="Location")
    ],
    outputs=gr.Text(label="Predicted Price"),
    title="🏡 House Price Predictor",
    description="Enter details to estimate house price (in ₹ lakhs). Uses Ridge Regression
)

# Launch app
interface.launch(share=True)
```

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarn
  fig.canvas.print_figure(bytes_io, **kw)
```







📊 Linear Regression Results:
MSE: 947.5
R² Score: -150.6

📊 Ridge Regression Results:
MSE: 623.71
R² Score: -98.79

📊 Lasso Regression Results:
MSE: 955.25
R² Score: -151.84

Colab notebook detected. To show errors in colab notebook, set debug=True in lau
* Running on public URL: https://39e06f7cd3976a8f9d.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades,