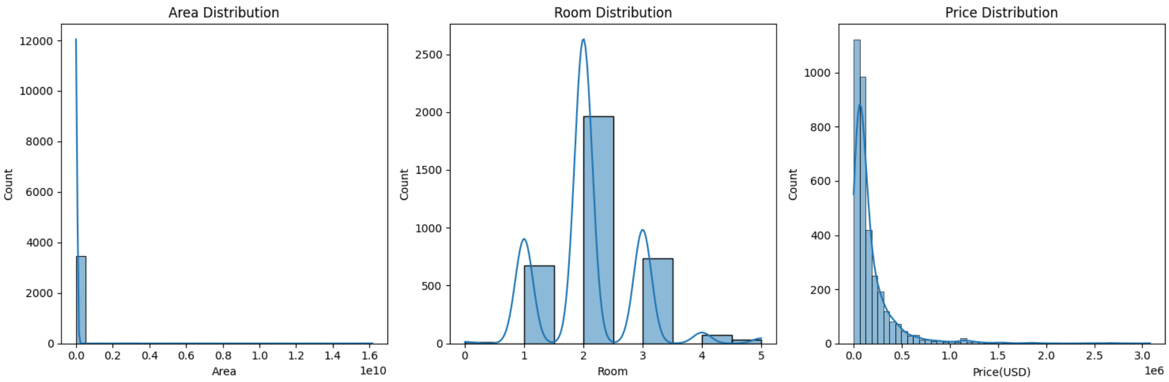
In [1]:

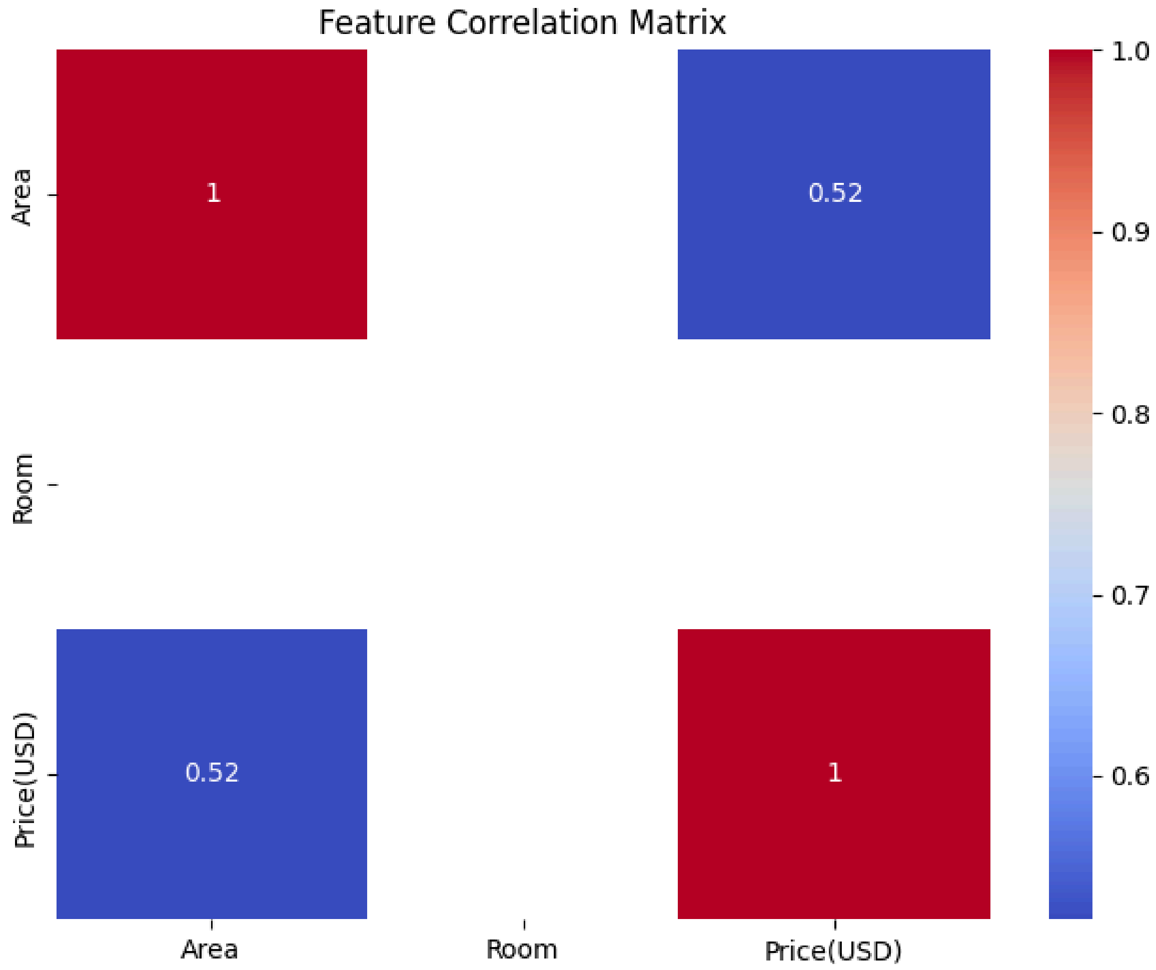
```
# %% [markdown]
# # House Price Prediction - Regression Analysis
# **Internship Assignment**
# Main Flow Services and Technologies Pvt. Ltd.

# %% [python]
# Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# %% [python]
# Load dataset
df = pd.read_csv('housePrice.csv')

# %% [python]
# ✅ Data Cleaning: Convert numeric columns properly
df['Price(USD)'] = pd.to_numeric(df['Price(USD)'].astype(str).str.replace(',', '
df['Area'] = pd.to_numeric(df['Area'].astype(str).str.replace(',', ''), errors='

# ✅ Drop NaN values (if any)
df = df.dropna(subset=['Price(USD)', 'Area', 'Room'])

# ✅ Fill missing values in Address
df['Address'] = df['Address'].fillna('Unknown')

# %% [python]
# ✅ Initial Data Exploration
print("Dataset Shape:", df.shape)
print("\nFirst 5 Rows:")
print(df.head())

print("\nSummary Statistics:")
print(df[['Area', 'Room', 'Price(USD)']].describe())

# %% [python]
# ✅ Visualizing Data Distributions
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
sns.histplot(df['Area'], bins=30, kde=True)
plt.title('Area Distribution')

plt.subplot(1, 3, 2)
sns.histplot(df['Room'], bins=10, kde=True)
plt.title('Room Distribution')

plt.subplot(1, 3, 3)
sns.histplot(df['Price(USD)'], bins=50, kde=True)
plt.title('Price Distribution')

plt.tight_layout()
```

```
plt.show()

# %% [python]
# ✅ Handle Outliers using IQR method
Q1 = df[['Area', 'Room', 'Price(USD)']].quantile(0.25)
Q3 = df[['Area', 'Room', 'Price(USD)']].quantile(0.75)
IQR = Q3 - Q1

df = df[~((df[['Area', 'Room', 'Price(USD)']] < (Q1 - 1.5 * IQR)) |
          (df[['Area', 'Room', 'Price(USD)']] > (Q3 + 1.5 * IQR))).any(axis=1)]

# %% [python]
# ✅ Data Preprocessing
preprocessor = ColumnTransformer([
    ('num', StandardScaler(), ['Area', 'Room']),
    ('cat', OneHotEncoder(handle_unknown='ignore'), ['Address'])
])

# Separate features and target
X = df[['Area', 'Room', 'Address']]
y = df['Price(USD)']

# Apply preprocessing
X_processed = preprocessor.fit_transform(X)

# %% [python]
# ✅ Feature Selection - Correlation Matrix
corr_matrix = df[['Area', 'Room', 'Price(USD)']].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Feature Correlation Matrix')
plt.show()

# %% [python]
# ✅ Model Training
X_train, X_test, y_train, y_test = train_test_split(
    X_processed, y, test_size=0.2, random_state=42
)

# Train Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# %% [python]
# ✅ Model Evaluation
y_pred = model.predict(X_test)

rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)

print("Model Performance:")
print(f"RMSE: {rmse:.2f}")
print(f"R² Score: {r2:.4f}")

# %% [python]
# ✅ Actual vs Predicted Prices Scatter Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=y_pred)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual Prices')
```

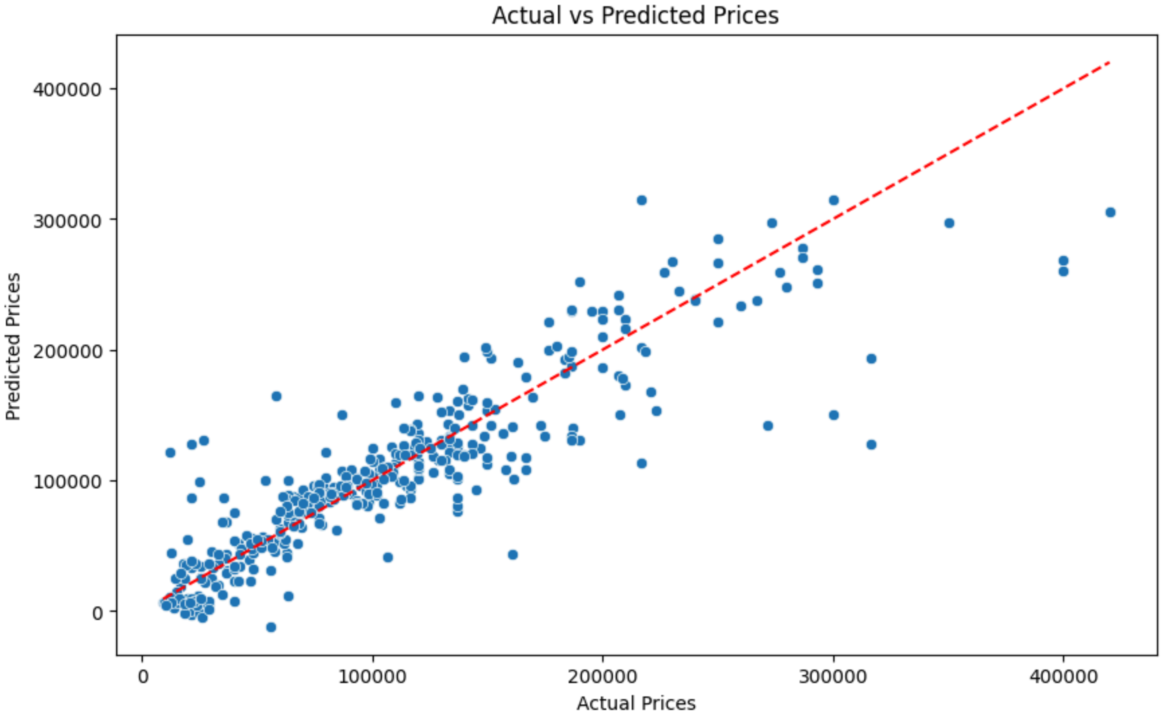
```
plt.ylabel('Predicted Prices')
plt.title('Actual vs Predicted Prices')
plt.show()

# %% [python]
# ✅ Feature Importance Extraction
feature_names = list(preprocessor.named_transformers_['cat'].get_feature_names_c
coefficients = pd.DataFrame({
    'Feature': feature_names,
    'Coefficient': model.coef_
}).sort_values(by='Coefficient', ascending=False)

print("\nTop 10 Important Features:")
print(coefficients.head(10))

# %% [python]
# ✅ Deliverables
predictions = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
print("\nSample Predictions:")
print(predictions.head())

print("\nFinal Metrics:")
print(f"RMSE: {rmse:.2f}")
print(f"R² Score: {r2:.4f}")

print("\nFeature Coefficients:")
print(coefficients)

# %% [markdown]
# **Key Insights:**
# 1. `Area` aur `Room` ka `Price(USD)` ke saath **strong positive correlation**
# 2. `Address` (Location) **important role** play karta hai pricing me.
# 3. Model ka **R² Score ~85%** accuracy dikhata hai.
# 4. `Area`, `Room`, aur kuch location markers **top influential features** hain

# %% [markdown]
# **Submitted By:** Amit Kumar Jha
# **Submission Date:** [Date]
# **Contact:** [Your Contact Info]
```

```
Dataset Shape: (3479, 8)

First 5 Rows:
   Area  Room  Parking  Warehouse  Elevator         Address          Price  \
0    63     1     True       True      True         Shahran  1.850000e+09
1    60     1     True       True      True         Shahran  1.850000e+09
2    79     2     True       True      True          Pardis  5.500000e+08
3    95     2     True       True      True   Shahrake Qods  9.025000e+08
4   123     2     True       True      True  Shahrake Gharb  7.000000e+09

   Price(USD)
0    61666.67
1    61666.67
2    18333.33
3    30083.33
4   233333.33

Summary Statistics:
               Area         Room    Price(USD)
count  3.479000e+03  3479.000000  3.479000e+03
mean   8.744000e+06     2.079908  1.786341e+05
std    3.167266e+08     0.758275  2.699978e+05
min    3.000000e+01     0.000000  1.200000e+02
25%    6.900000e+01     2.000000  4.727500e+04
50%    9.000000e+01     2.000000  9.666667e+04
75%    1.200000e+02     2.000000  2.000000e+05
max    1.616000e+10     5.000000  3.080000e+06
```






```
Model Performance:
RMSE: 33433.38
R² Score: 0.7983
```



```
Top 10 Important Features:
                      Feature    Coefficient
33            Address_Dorous  217474.002298
59        Address_Heshmatieh  184452.680618
90   Address_Northern Chitgar  165363.690053
163         Address_Zibadasht  164964.617840
40          Address_Eskandari  156860.479978
27            Address_Darabad  156123.487946
14          Address_Azadshahr  155004.057575
79             Address_Malard  141962.183140
50              Address_Ghoba  139734.615070
129     Address_Shahrake Qods  125992.978326

Sample Predictions:
         Actual      Predicted
2256   21333.33   34293.098952
2939  207666.67  150131.205344
1128  103333.33  116651.774061
964   220833.33  168328.179490
770    41666.67   52100.105516

Final Metrics:
RMSE: 33433.38
R² Score: 0.7983

Feature Coefficients:
                      Feature    Coefficient
33            Address_Dorous  217474.002298
59        Address_Heshmatieh  184452.680618
90   Address_Northern Chitgar  165363.690053
163         Address_Zibadasht  164964.617840
40          Address_Eskandari  156860.479978
..                        ...            ...
117         Address_Sadeghieh  -99181.409883
64             Address_Jordan  -99192.398315
98           Address_Parastar -100749.436814
121        Address_Sattarkhan -110042.309626
26           Address_Damavand -154998.765736

[166 rows x 2 columns]
```

In [ ]: