```
# Import libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score



# Load dataset
df = pd.read_csv("/content/Housing.csv")



# Check the first few rows
print(df.head())
```

```
       price  area  bedrooms  bathrooms  stories mainroad guestroom basement  \
0  13300000  7420         4          2        3      yes        no       no
1  12250000  8960         4          4        4      yes        no       no
2  12250000  9960         3          2        2      yes        no      yes
3  12215000  7500         4          2        2      yes        no      yes
4  11410000  7420         4          1        2      yes       yes      yes

  hotwaterheating airconditioning  parking prefarea furnishingstatus
0              no             yes        2      yes        furnished
1              no             yes        3       no        furnished
2              no              no        2      yes   semi-furnished
3              no             yes        3      yes        furnished
4              no             yes        2       no        furnished
```

```
# Encode categorical variables if any
df_encoded = pd.get_dummies(df, drop_first=True)


# Separate features (X) and target (y)
# Assuming 'price' is the target variable
X = df_encoded.drop("price", axis=1)
y = df_encoded["price"]



# Split dataset into train & test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Create and fit the model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

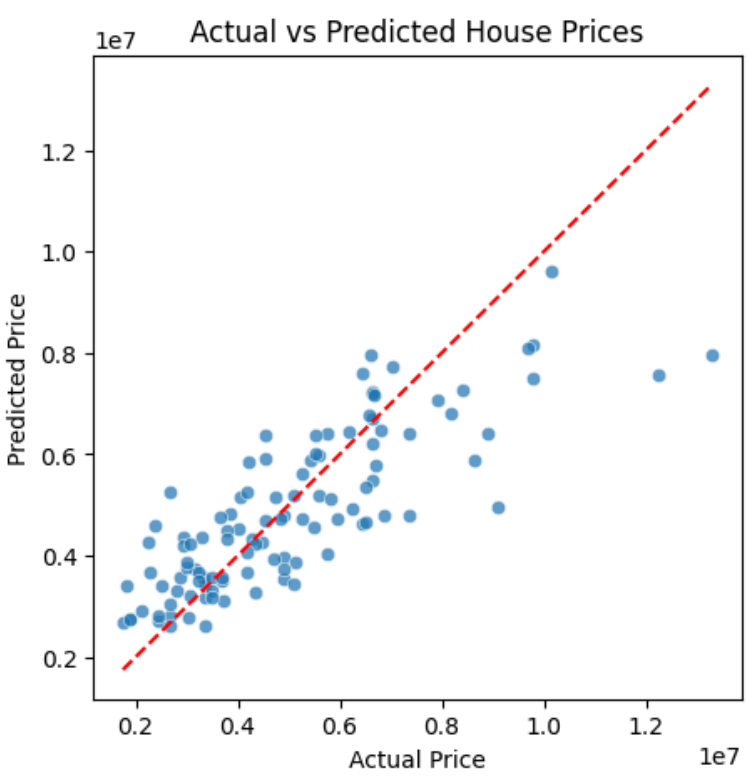
```
# Predictions
y_pred = model.predict(X_test)



# Evaluation
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
```

```
print("Model Coefficients:", model.coef_)
print("Intercept:", model.intercept_)
print("Root Mean Squared Error (RMSE):", rmse)
print("R² Score:", r2)
```

```
Model Coefficients: [ 2.35968805e+02  7.67787016e+04  1.09444479e+06  4.07476595e+05
  2.24841913e+05  3.67919948e+05  2.31610037e+05  3.90251176e+05
  6.84649885e+05  7.91426736e+05  6.29890565e+05 -1.26881818e+05
 -4.13645062e+05]
Intercept: 260032.35760741215
Root Mean Squared Error (RMSE): 1324506.9600914386
R² Score: 0.6529242642153184
```

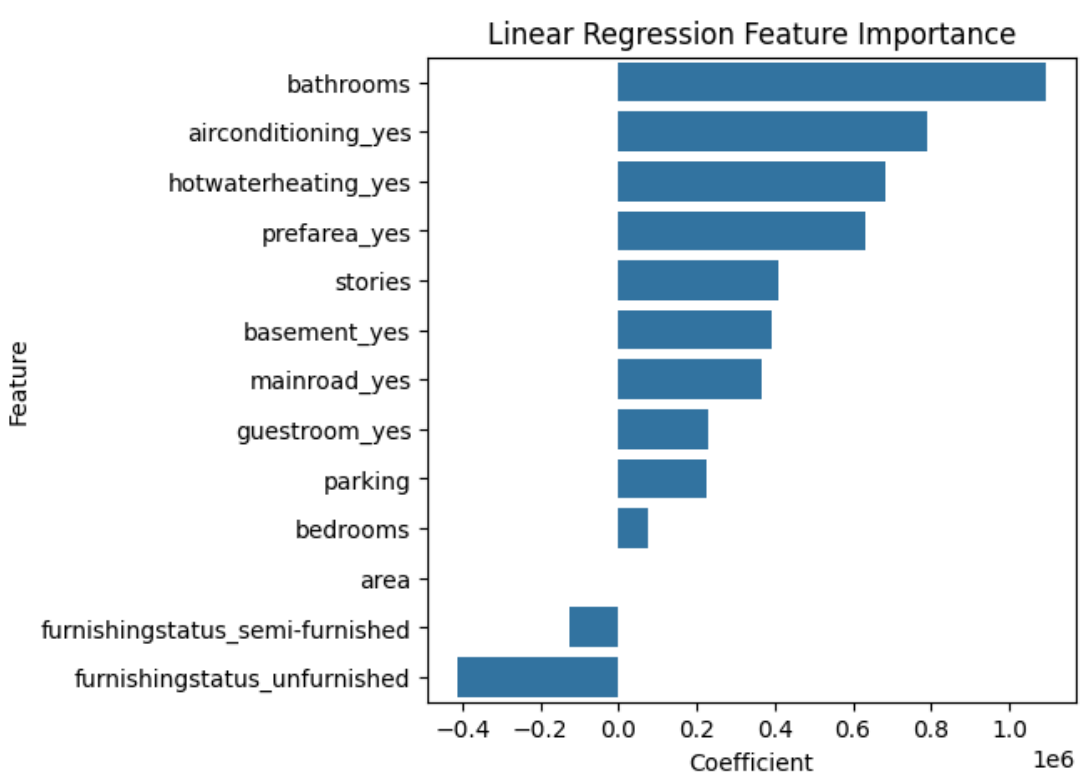
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
#Visualization - Actual vs Predicted
plt.figure(figsize=(5,5))
sns.scatterplot(x=y_test, y=y_pred, alpha=0.7)
plt.xlabel("Actual Price")
plt.ylabel("Predicted Price")
plt.title("Actual vs Predicted House Prices")
plt.plot([y.min(), y.max()], [y.min(), y.max()], color="red", linestyle="--")
plt.show()
```

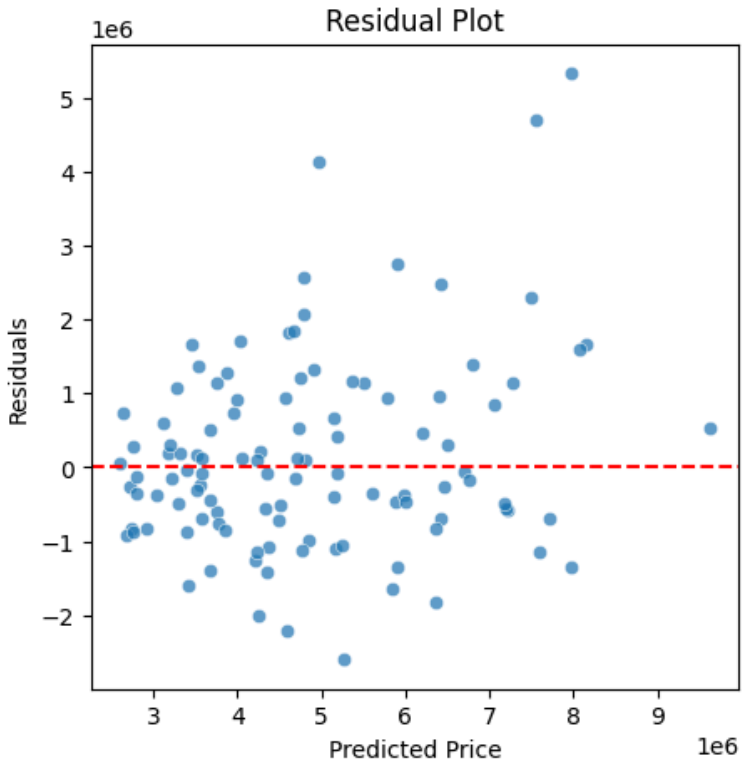



```
# Visualization - Feature importance
coef_df = pd.DataFrame({
    "Feature": X.columns,
    "Coefficient": model.coef_
}).sort_values(by="Coefficient", ascending=False)
```

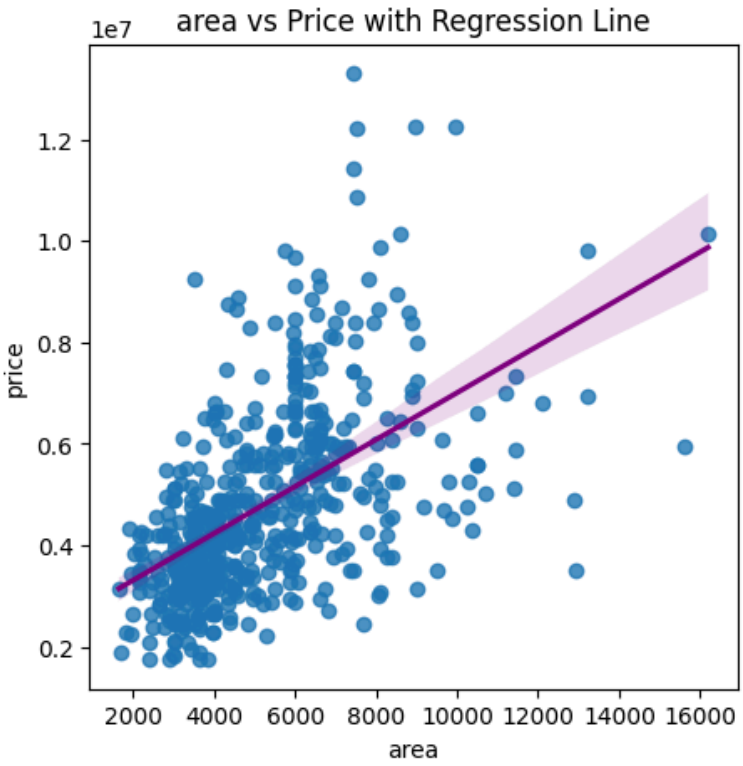
```
plt.figure(figsize=(5,5))
sns.barplot(x="Coefficient", y="Feature", data=coef_df)
plt.title("Linear Regression Feature Importance")
plt.show()
```

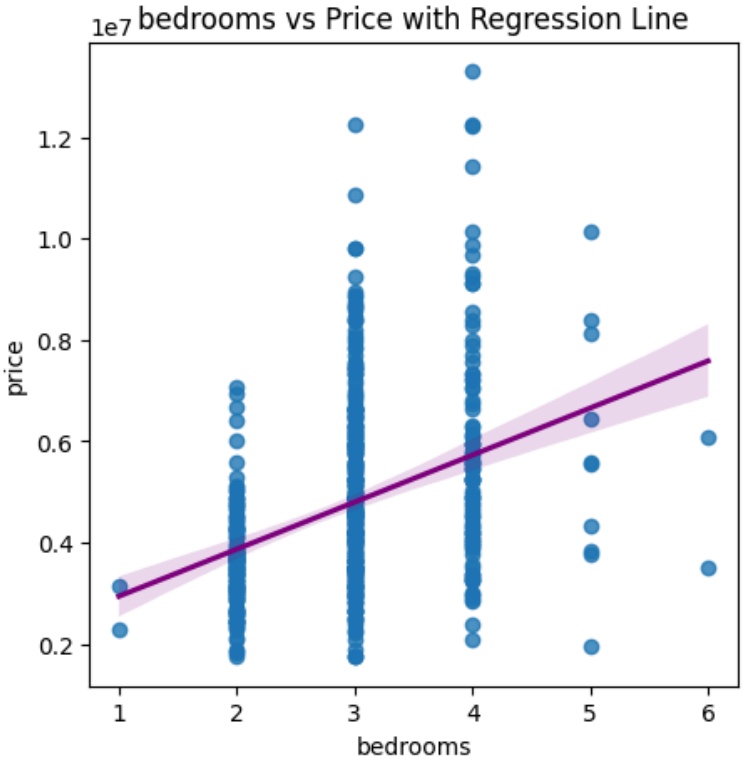



```
#Visualization - Residual Plot
residuals = y_test - y_pred
plt.figure(figsize=(5,5))
sns.scatterplot(x=y_pred, y=residuals, alpha=0.7)
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel("Predicted Price")
plt.ylabel("Residuals")
plt.title("Residual Plot")
plt.show()
```

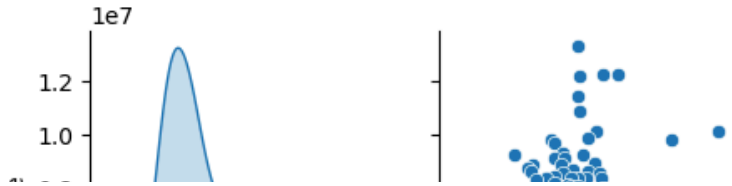



```
# Scatter plots with regression line
for feature in ['area', 'bedrooms']:
    plt.figure(figsize=(5,5))
    sns.regplot(x=df[feature], y=df['price'], line_kws={"color": "purple"})
    plt.title(f"{feature} vs Price with Regression Line")
    plt.show()
```






```
# Pair plot (selected features + target)
selected_cols = ['price', 'area']
sns.pairplot(df[selected_cols], diag_kind='kde')
plt.show()
```


1e7
1.2
1.0