In [1]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in /home/anaconda/anaconda3/lib/pytho
n3.12/site-packages (1.26.4)
Could not fetch URL https://pypi.org/simple/pip/: There was a problem conf
irming the ssl certificate: HTTPSConnectionPool(host='pypi.org', port=44
3): Max retries exceeded with url: /simple/pip/ (Caused by SSLError(SSLCer
tVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify
failed: unable to get local issuer certificate (_ssl.c:1000)'))) - skippin
g
Note: you may need to restart the kernel to use updated packages.
```

In [ ]:
```
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linearfrom sklearn.model_selection import train_test_split
_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [5]:
```
df = pd.read_csv('doc.csv')
```

In [6]:
```
df
```

Out[6]:

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	
0	3	2	1500	5000	1	0	0	3	
1	4	3	2500	6000	2	1	1	4	
2	2	1	900	3000	1	0	0	3	
3	5	4	3200	8000	2	1	2	5	
4	3	2	1800	4000	1	0	1	4	
...	...	...	...	...	...	...	...	...	
91	2	1	950	3200	1	0	0	3	
92	5	4	3300	8200	2	1	2	5	
93	3	2	1900	4200	1	0	1	4	
94	4	3	2500	6800	2	1	1	4	
95	2	1	1100	3600	1	0	0	3	

96 rows × 9 columns

In [7]:
```
df.describe()
```

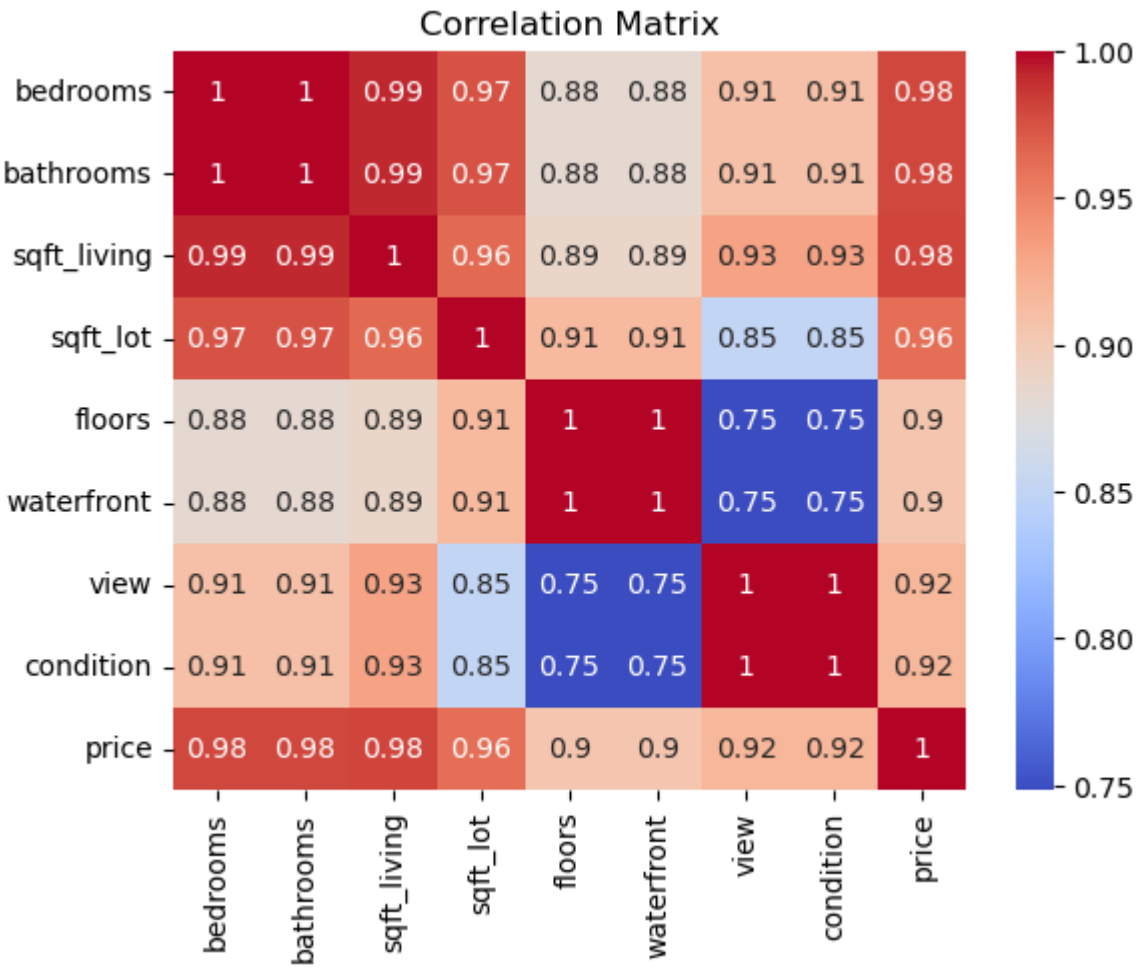
Out[7]:

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	
count	96.000000	96.000000	96.000000	96.000000	96.000000	96.000000	96.00
mean	3.375000	2.375000	2003.125000	5362.500000	1.437500	0.437500	0.81
std	1.058798	1.058798	796.003587	1727.623863	0.498682	0.498682	0.72
min	2.000000	1.000000	900.000000	3000.000000	1.000000	0.000000	0.00
25%	2.750000	1.750000	1400.000000	3900.000000	1.000000	0.000000	0.00
50%	3.000000	2.000000	1850.000000	4900.000000	1.000000	0.000000	1.00
75%	4.000000	3.000000	2525.000000	6850.000000	2.000000	1.000000	1.00
max	5.000000	4.000000	3300.000000	8200.000000	2.000000	1.000000	2.00

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
bedrooms       0
bathrooms      0
sqft_living    0
sqft_lot       0
floors         0
waterfront     0
view           0
condition      0
price          0
dtype: int64
```

In [10]:
```
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```



In [12]:
```
X = df[['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'wa
y = df['price']
```

In [13]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
```

In [14]:
```
# Building the Linear Regression Model
model = LinearRegression()
# Fitting the model on the training data
model.fit(X_train, y_train)
```

Out[14]:
```
▾  LinearRegression i ?

LinearRegression()
```

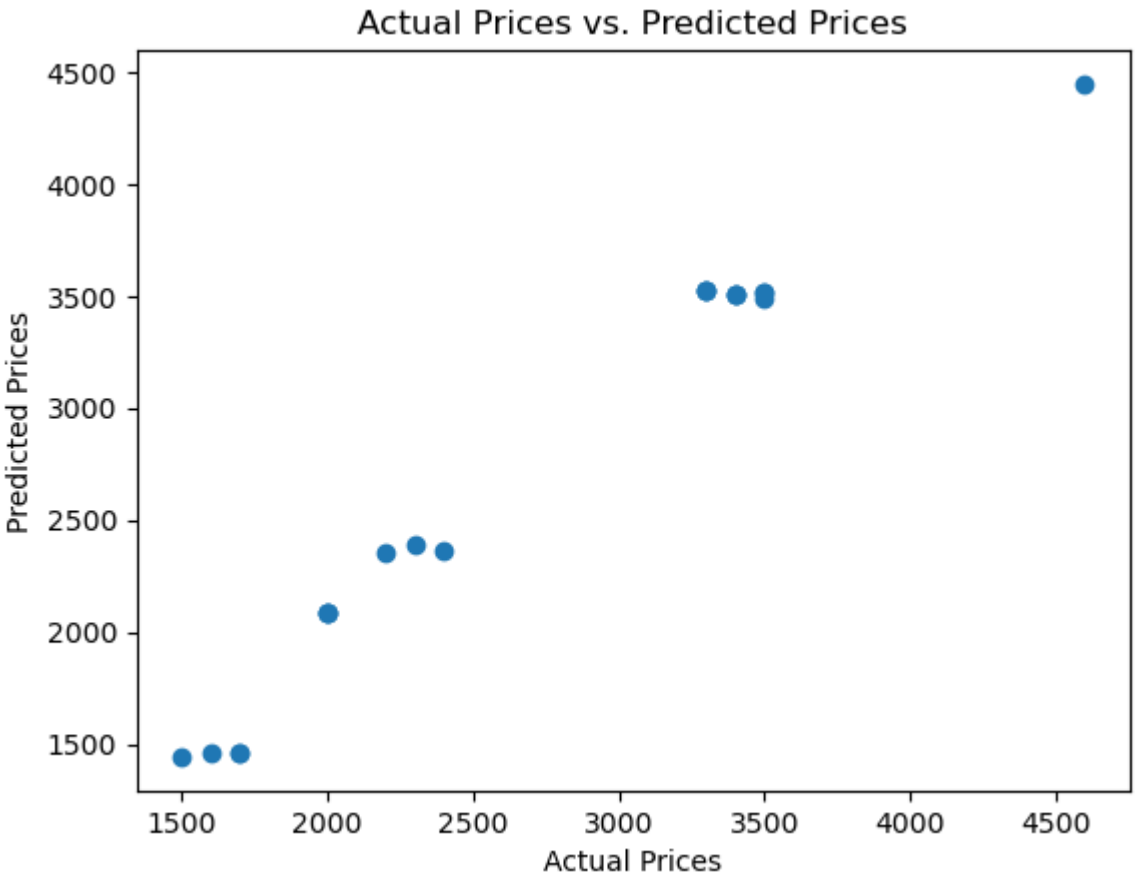
In [16]:
```
y_pred = model.predict(X_test)
```

In [17]:
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

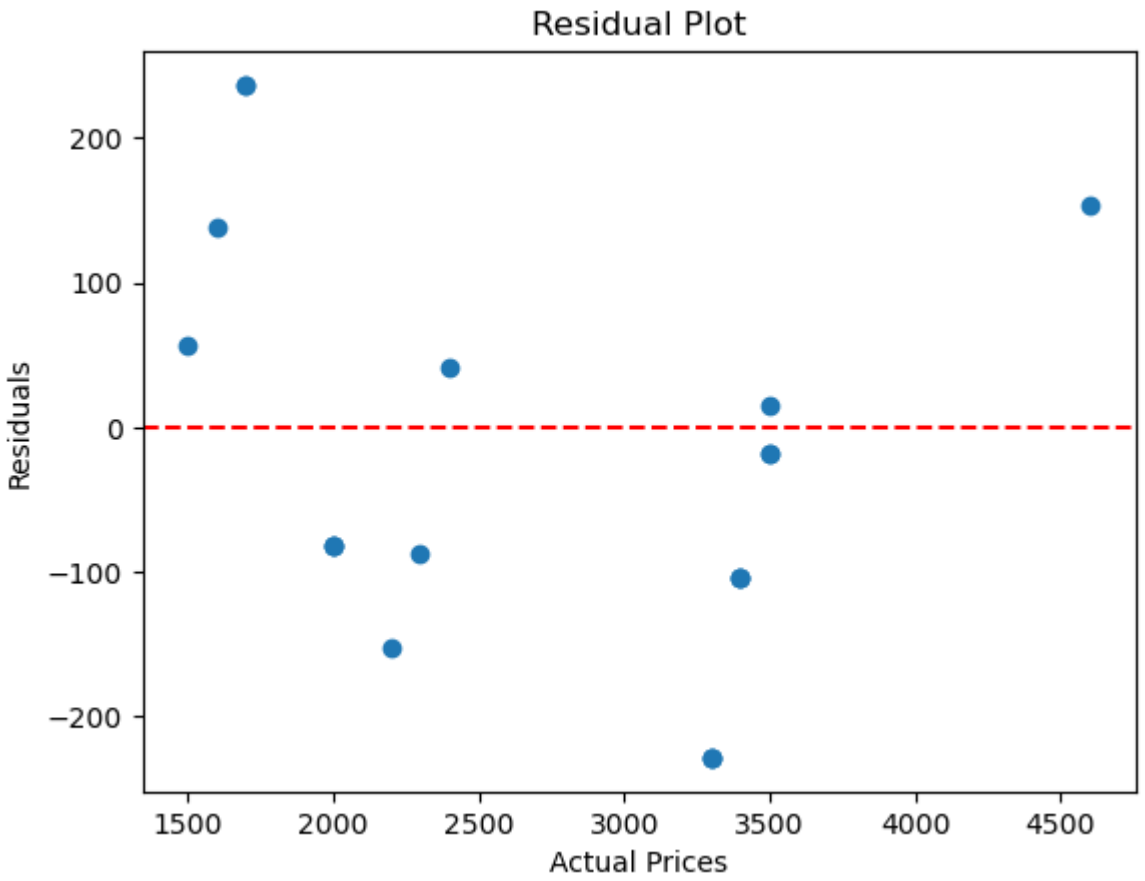
```
Mean Squared Error: 20027.086153182714
R-squared: 0.9727188582574816
```

In [19]:
```
# Predictions and Visualization
# To visualize the predictions against actual prices, we'll use a scatter
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Prices")
```

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



In [20]:
```
# We can also create a residual plot to check the model's performance
residuals = y_test - y_pred
plt.scatter(y_test, residuals)
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel("Actual Prices")
plt.ylabel("Residuals")
plt.title("Residual Plot")
plt.show()
```



In [21]:
```
new_data = [[3, 2, 1500, 4000, 1, 0, 0, 3]]
predicted_price = model.predict(new_data)
print("Predicted Price:", predicted_price[0])
```

```
Predicted Price: 2040.681186468728
/home/anaconda/anaconda3/lib/python3.12/site-packages/sklearn/base.py:493:
UserWarning: X does not have valid feature names, but LinearRegression was
fitted with feature names
  warnings.warn(
```

In [ ]: