```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import drive

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
```

```
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
df=pd.read_csv("/content/drive/MyDrive/python-Saylani/Advertising Budget and Sales.csv")
df.head()
```

	Unnamed: 0	TV Ad Budget ($)	Radio Ad Budget ($)	Newspaper Ad Budget ($)	Sales ($)
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

```
df.shape
```

```
(200, 5)
```

```
df.isnull().sum()
```

	0
Unnamed: 0	0
TV Ad Budget ($)	0
Radio Ad Budget ($)	0
Newspaper Ad Budget ($)	0
Sales ($)	0

dtype: int64

```
df.duplicated().sum()
```

```
np.int64(0)
```

```
df.drop('Unnamed: 0', axis=1,inplace=True)
df.head()
```

	TV Ad Budget ($)	Radio Ad Budget ($)	Newspaper Ad Budget ($)	Sales ($)
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

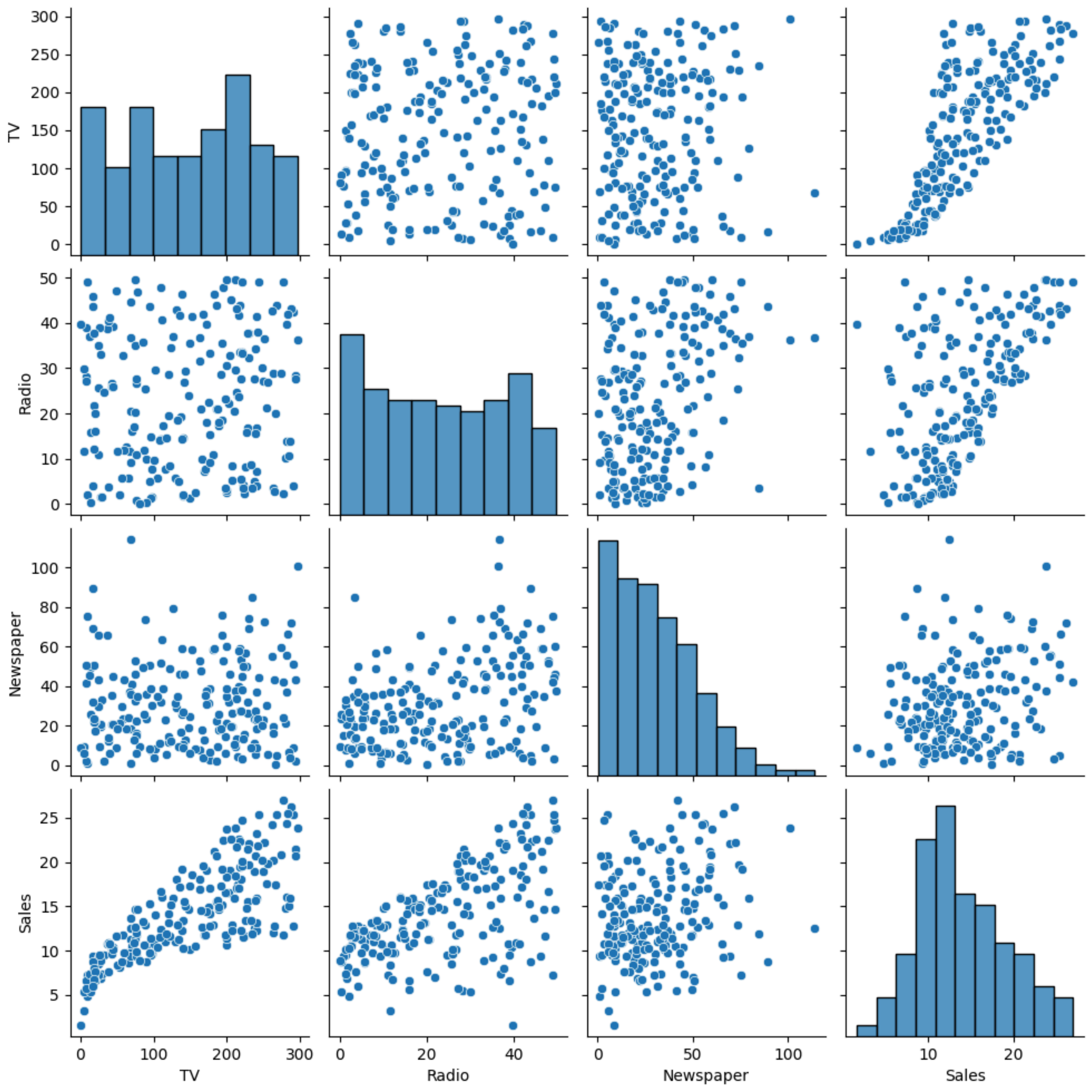
```
df.shape
```

```
(200, 4)
```

```
df = df.rename(columns={'TV Ad Budget ($)':'TV',    'Radio Ad Budget ($)':'Radio',  'Newspaper Ad Budget ($)':'Newspaper',  'Sa
df.head()
```

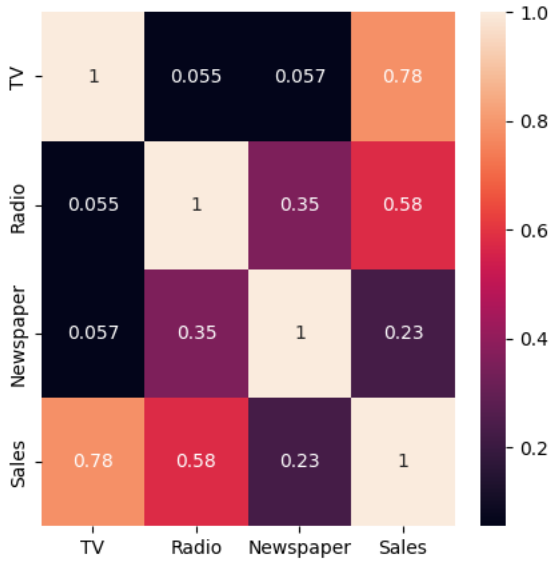
	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

```
sns.pairplot(df)
plt.show()
```

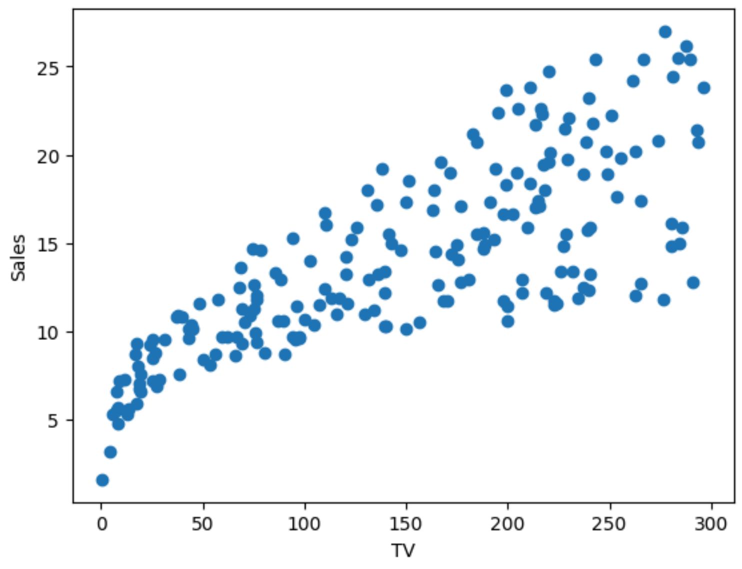


```
plt.figure(figsize=(5,5))
sns.heatmap(df.corr(),annot=True)
```

```
<Axes: >
```



```
plt.scatter(df['TV'], df['Sales'])
plt.xlabel('TV')
plt.ylabel('Sales')
plt.show()
```



```
x=df['TV']
y=df['Sales']
```

```
x.shape
```

```
(200,)
```

```
x=x.values.reshape(200,1)
x.shape
```

```
(200, 1)
```

```
y.shape
```

```
(200,)
```

```
y=y.values.reshape(200,1)
y.shape
```

```
(200, 1)
```

```
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)
```

```
from sklearn.linear_model import LinearRegression
```

```
lr=LinearRegression()
```

Simple Linear Regression

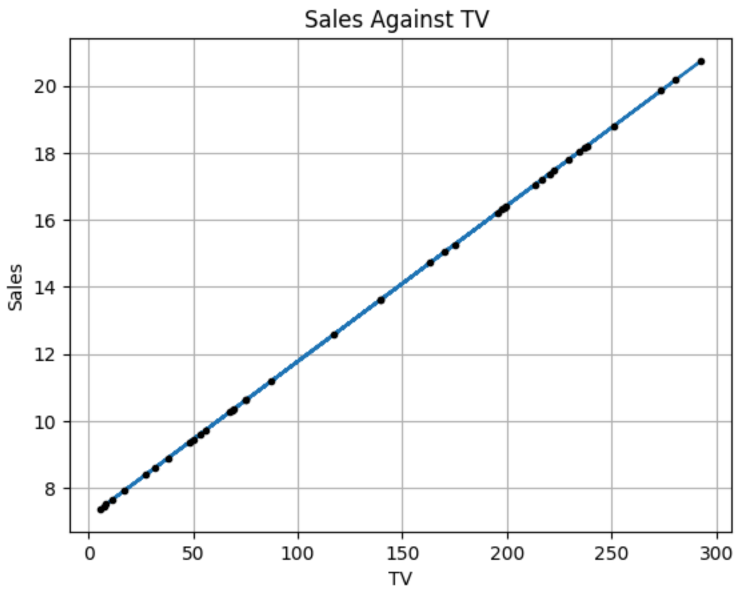
```
lr.fit(x_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
y_pred=lr.predict(x_test)
y_pred
```

```
array([[14.71794394],
       [16.2115484 ],
       [20.74819743],
       [ 7.66403631],
       [17.37013877],
       [10.61402143],
       [17.2072847 ],
       [ 9.44612512],
       [17.46785121],
       [15.2669948 ],
       [ 8.58532504],
       [ 9.73460946],
       [18.03086098],
       [ 7.37089899],
       [13.61053628],
       [15.03899911],
       [ 7.45930549],
       [16.31391381],
       [10.62332738],
       [18.16579721],
       [17.79821232],
       [10.27435438],
       [ 8.88776831],
       [18.79394862],
       [10.33019006],
       [ 9.60897918],
       [17.05373658],
       [13.60123034],
       [11.17703121],
       [ 7.51048819],
       [16.41627922],
       [10.339496  ],
       [16.37440246],
       [ 7.90599093],
       [20.18053468],
       [18.203021  ],
       [ 9.36702457],
       [19.85482655],
       [12.57292322],
       [ 8.39920611]])
```

```
plt.figure()
plt.title('Sales Against TV ')
plt.xlabel('TV')
plt.ylabel('Sales')
plt.plot(x_test,y_pred)
plt.plot(x_test,y_pred,'k.')
plt.grid()
plt.show()
```



```
for i, predictions in enumerate(y_pred):
    print('actual Price: %s   and    Predicted Price: %s' %(y_test[i],predictions))
```

```
actual Price: [16.9]   and    Predicted Price: [14.71794394]
actual Price: [22.4]   and    Predicted Price: [16.2115484]
actual Price: [21.4]   and    Predicted Price: [20.74819743]
actual Price: [7.3]   and    Predicted Price: [7.66403631]
actual Price: [24.7]   and    Predicted Price: [17.37013877]
actual Price: [12.6]   and    Predicted Price: [10.61402143]
actual Price: [22.3]   and    Predicted Price: [17.2072847]
actual Price: [8.4]   and    Predicted Price: [9.44612512]
actual Price: [11.5]   and    Predicted Price: [17.46785121]
actual Price: [14.9]   and    Predicted Price: [15.2669948]
actual Price: [9.5]   and    Predicted Price: [8.58532504]
actual Price: [8.7]   and    Predicted Price: [9.73460946]
actual Price: [11.9]   and    Predicted Price: [18.03086098]
actual Price: [5.3]   and    Predicted Price: [7.37089899]
actual Price: [10.3]   and    Predicted Price: [13.61053628]
actual Price: [11.7]   and    Predicted Price: [15.03899911]
actual Price: [5.5]   and    Predicted Price: [7.45930549]
actual Price: [16.6]   and    Predicted Price: [16.31391381]
actual Price: [11.3]   and    Predicted Price: [10.62332738]
actual Price: [18.9]   and    Predicted Price: [18.16579721]
actual Price: [19.7]   and    Predicted Price: [17.79821232]
actual Price: [12.5]   and    Predicted Price: [10.27435438]
actual Price: [10.9]   and    Predicted Price: [8.88776831]
actual Price: [22.2]   and    Predicted Price: [18.79394862]
actual Price: [9.3]   and    Predicted Price: [10.33019006]
actual Price: [8.1]   and    Predicted Price: [9.60897918]
actual Price: [21.7]   and    Predicted Price: [17.05373658]
actual Price: [13.4]   and    Predicted Price: [13.60123034]
actual Price: [10.6]   and    Predicted Price: [11.17703121]
actual Price: [5.7]   and    Predicted Price: [7.51048819]
actual Price: [10.6]   and    Predicted Price: [16.41627922]
actual Price: [11.3]   and    Predicted Price: [10.339496]
actual Price: [23.7]   and    Predicted Price: [16.37440246]
actual Price: [8.7]   and    Predicted Price: [7.90599093]
actual Price: [16.1]   and    Predicted Price: [20.18053468]
actual Price: [20.7]   and    Predicted Price: [18.203021]
actual Price: [11.6]   and    Predicted Price: [9.36702457]
actual Price: [20.8]   and    Predicted Price: [19.85482655]
actual Price: [11.9]   and    Predicted Price: [12.57292322]
actual Price: [6.9]   and    Predicted Price: [8.39920611]
```

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

```
# y_test = actual values
# y_pred = model predictions

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
```

```
r2 = r2_score(y_test, y_pred)

print("MAE :", mae)
print("MSE :", mse)
print("RMSE:", rmse)
print("R2 Score:", r2)
```

```
MAE : 2.444420003751042
MSE : 10.204654118800956
RMSE: 3.194472431998898
R2 Score: 0.6766954295627076
```

Multi Linear Regresion

```
x= df.drop('Sales', axis=1)
y= df['Sales']
```

```
x
```

	TV	Radio	Newspaper
0	230.1	37.8	69.2
1	44.5	39.3	45.1
2	17.2	45.9	69.3
3	151.5	41.3	58.5
4	180.8	10.8	58.4
...	...	...	...
195	38.2	3.7	13.8
196	94.2	4.9	8.1
197	177.0	9.3	6.4
198	283.6	42.0	66.2
199	232.1	8.6	8.7

200 rows × 3 columns

```
y
```

	Sales
0	22.1
1	10.4
2	9.3
3	18.5
4	12.9
...	...
195	7.6
196	9.7
197	12.8
198	25.5
199	13.4

200 rows × 1 columns

dtype: float64

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

```
lr.fit(x_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
y_pred=lr.predict(x_test)
y_pred
```

```
array([16.4080242 , 20.88988209, 21.55384318, 10.60850256, 22.11237326,
       13.10559172, 21.05719192,  7.46101034, 13.60634581, 15.15506967,
        9.04831992,  6.65328312, 14.34554487,  8.90349333,  9.68959028,
       12.16494386,  8.73628397, 16.26507258, 10.27759582, 18.83109103,
       19.56036653, 13.25103464, 12.33620695, 21.30695132,  7.82740305,
        5.80957448, 20.75753231, 11.98138077,  9.18349576,  8.5066991 ,
       12.46646769, 10.00337695, 21.3876709 , 12.24966368, 18.26661538,
       20.13766267, 14.05514005, 20.85411186, 11.0174441 ,  4.56899622])
```

```
for i, predictions in enumerate(y_pred):
    print('actual Price: %s   and    Predicted Price: %s' %(y_test.iloc[i],predictions))
```

```
actual Price: 16.9   and    Predicted Price: 16.408024203228628
actual Price: 22.4   and    Predicted Price: 20.889882087147885
actual Price: 21.4   and    Predicted Price: 21.553843179089558
actual Price: 7.3   and    Predicted Price: 10.6085025619849
actual Price: 24.7   and    Predicted Price: 22.112373259857662
actual Price: 12.6   and    Predicted Price: 13.105591724016454
actual Price: 22.3   and    Predicted Price: 21.057191916314647
actual Price: 8.4   and    Predicted Price: 7.461010344558369
actual Price: 11.5   and    Predicted Price: 13.60634580543393
actual Price: 14.9   and    Predicted Price: 15.155069668921398
actual Price: 9.5   and    Predicted Price: 9.048319924103865
actual Price: 8.7   and    Predicted Price: 6.653283124939039
actual Price: 11.9   and    Predicted Price: 14.345544865081424
actual Price: 5.3   and    Predicted Price: 8.903493328870406
actual Price: 10.3   and    Predicted Price: 9.689590280381115
actual Price: 11.7   and    Predicted Price: 12.16494385914241
actual Price: 5.5   and    Predicted Price: 8.736283973362841
actual Price: 16.6   and    Predicted Price: 16.265072577249363
actual Price: 11.3   and    Predicted Price: 10.277595820569246
actual Price: 18.9   and    Predicted Price: 18.83109103439628
actual Price: 19.7   and    Predicted Price: 19.560366533092832
actual Price: 12.5   and    Predicted Price: 13.251034642395638
actual Price: 10.9   and    Predicted Price: 12.336206948241493
actual Price: 22.2   and    Predicted Price: 21.306951317459145
actual Price: 9.3   and    Predicted Price: 7.827403050750988
actual Price: 8.1   and    Predicted Price: 5.809574478072964
actual Price: 21.7   and    Predicted Price: 20.757532314509962
actual Price: 13.4   and    Predicted Price: 11.981380774195218
actual Price: 10.6   and    Predicted Price: 9.183495762801746
actual Price: 5.7   and    Predicted Price: 8.506699100151758
actual Price: 10.6   and    Predicted Price: 12.466467693418515
actual Price: 11.3   and    Predicted Price: 10.003376951209537
actual Price: 23.7   and    Predicted Price: 21.38767090231054
actual Price: 8.7   and    Predicted Price: 12.249663675504417
actual Price: 16.1   and    Predicted Price: 18.26661537607972
actual Price: 20.7   and    Predicted Price: 20.137662665419217
actual Price: 11.6   and    Predicted Price: 14.055140052778944
actual Price: 20.8   and    Predicted Price: 20.85411186286339
actual Price: 11.9   and    Predicted Price: 11.0174441001449
actual Price: 6.9   and    Predicted Price: 4.568996222153632
```

```
# y_test = actual values
# y_pred = model predictions

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

print("MAE :", mae)
print("MSE :", mse)
print("RMSE:", rmse)
print("R2 Score:", r2)
```

```
MAE : 1.4607567168117603
MSE : 3.1740973539761033
RMSE: 1.78159966153345
R2 Score: 0.899438024100912
```

Polynomial Regression

```
poly_feature=PolynomialFeatures(degree=2)
```

```
x_train=poly_feature.fit_transform(x_train)
x_test=poly_feature.fit_transform(x_test)
```

```
pr=LinearRegression()
```

```
pr.fit(x_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
y_pred=pr.predict(x_test)
y_pred
```

```
array([17.25443578, 22.7193321 , 20.42799436,  7.542709  , 24.37103037,
       12.55927164, 22.80784986,  8.34371358, 12.0586023 , 15.69024565,
        7.89166367,  8.27082715, 11.86055971,  6.16591094, 10.57664393,
       12.3628954 ,  6.74390963, 16.65082111, 10.68661722, 19.03888161,
       20.15223945, 13.1137814 ,  9.56273868, 22.10675225,  8.96725241,
        7.7794437 , 22.40745151, 12.72065973, 10.25529986,  6.22368636,
       11.64677688, 10.22431946, 23.39763949,  9.17403232, 15.36143449,
       21.05675814, 10.9795286 , 20.23370753, 11.85153256,  6.58779915])
```

```
for i, predictions in enumerate(y_pred):
    print('actual Price: %s   and    Predicted Price: %s' %(y_test.iloc[i],predictions))
```

```
actual Price: 16.9   and    Predicted Price: 17.25443578421151
actual Price: 22.4   and    Predicted Price: 22.719332098787888
actual Price: 21.4   and    Predicted Price: 20.4279943599341
actual Price: 7.3   and    Predicted Price: 7.542708996441649
actual Price: 24.7   and    Predicted Price: 24.371030368769677
actual Price: 12.6   and    Predicted Price: 12.559271635604164
actual Price: 22.3   and    Predicted Price: 22.807849861474054
actual Price: 8.4   and    Predicted Price: 8.343713576679239
actual Price: 11.5   and    Predicted Price: 12.058602302533435
actual Price: 14.9   and    Predicted Price: 15.690245648360365
actual Price: 9.5   and    Predicted Price: 7.891663667904025
actual Price: 8.7   and    Predicted Price: 8.270827153870238
actual Price: 11.9   and    Predicted Price: 11.860559712580104
actual Price: 5.3   and    Predicted Price: 6.165910943379144
actual Price: 10.3   and    Predicted Price: 10.576643926736509
actual Price: 11.7   and    Predicted Price: 12.362895400851478
actual Price: 5.5   and    Predicted Price: 6.743909627103889
actual Price: 16.6   and    Predicted Price: 16.65082111309836
actual Price: 11.3   and    Predicted Price: 10.686617218701066
actual Price: 18.9   and    Predicted Price: 19.038881612147414
actual Price: 19.7   and    Predicted Price: 20.15223944894774
actual Price: 12.5   and    Predicted Price: 13.113781404373553
actual Price: 10.9   and    Predicted Price: 9.562738675620775
actual Price: 22.2   and    Predicted Price: 22.10675224573791
actual Price: 9.3   and    Predicted Price: 8.967252407306702
actual Price: 8.1   and    Predicted Price: 7.779443703471403
actual Price: 21.7   and    Predicted Price: 22.4074515071566
actual Price: 13.4   and    Predicted Price: 12.720659732639861
actual Price: 10.6   and    Predicted Price: 10.255299857895459
actual Price: 5.7   and    Predicted Price: 6.223686359442547
actual Price: 10.6   and    Predicted Price: 11.646776880239708
actual Price: 11.3   and    Predicted Price: 10.224319464837755
actual Price: 23.7   and    Predicted Price: 23.39763948689052
actual Price: 8.7   and    Predicted Price: 9.17403231875765
actual Price: 16.1   and    Predicted Price: 15.361434493590654
actual Price: 20.7   and    Predicted Price: 21.05675813802423
actual Price: 11.6   and    Predicted Price: 10.979528601220524
actual Price: 20.8   and    Predicted Price: 20.23370752759142
actual Price: 11.9   and    Predicted Price: 11.851532556115814
actual Price: 6.9   and    Predicted Price: 6.587799148930967
```

```
# y_test = actual values
# y_pred = model predictions

mae = mean_absolute_error(y_test, y_pred)
```

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

print("MAE :", mae)
print("MSE :", mse)
print("RMSE:", rmse)
print("R2 Score:", r2)
```

```
MAE : 0.5261794444043838
MSE : 0.41291022853790765
RMSE: 0.6425809120553673
R2 Score: 0.9869181490609602
```

```
Start coding or generate with AI.
```