In [1]:

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
data=pd.read_csv(r"C:\Users\gunis\Downloads\archive.zip")
```

In [3]:

```
data
```

Out[3]:

	Unnamed: 0.1	Unnamed: 0	brand	name	price	spec_rating	processor	CPU	Ram
0	0	0	HP	Victus 15-fb0157AX Gaming Laptop	49900	73.000000	5th Gen AMD Ryzen 5 5600H	Hexa Core, 12 Threads	8GB
1	1	1	HP	15s-fq5007TU Laptop	39900	60.000000	12th Gen Intel Core i3 1215U	Hexa Core (2P + 4E), 8 Threads	8GB
2	2	2	Acer	One 14 Z8-415 Laptop	26990	69.323529	11th Gen Intel Core i3 1115G4	Dual Core, 4 Threads	8GB
3	3	3	Lenovo	Yoga Slim 6 14IAP8 82WU0095IN Laptop	59729	66.000000	12th Gen Intel Core i5 1240P	12 Cores (4P + 8E), 16 Threads	16GB
4	4	4	Apple	MacBook Air 2020 MGND3HN Laptop	69990	69.323529	Apple M1	Octa Core (4P + 4E)	8GB
...	...	...	...	...	...	...	...	...	...
888	926	1015	Asus	Vivobook 15X 2023 K3504VAB-NJ321WS Laptop	44990	69.323529	13th Gen Intel Core i3 1315U	Hexa Core (2P + 4E), 8 Threads	8GB
889	927	1016	Asus	TUF A15 FA577RM-HQ032WS Laptop	110000	71.000000	6th Gen AMD Ryzen 7 6800H	Octa Core, 16 Threads	16GB
890	928	1017	Asus	ROG Zephyrus G14 2023 GA402XV-N2034WS Gaming L...	189990	89.000000	7th Gen AMD Ryzen 9 7940HS	Octa Core, 16 Threads	32GB
891	929	1018	Asus	TUF Gaming F15 2023 FX507VU-LP083WS Gaming Laptop	129990	73.000000	13th Gen Intel Core i7 13700H	14 Cores (6P + 8E), 20 Threads	16GB
892	930	1019	Asus	TUF Gaming A15 2023 FA577XU-LP041WS Gaming Laptop	131990	84.000000	7th Gen AMD Ryzen 9 7940HS	Octa Core, 16 Threads	16GB

893 rows × 18 columns

In [4]:
```
data.head()
```

Out[4]:

	Unnamed: 0.1	Unnamed: 0	brand	name	price	spec_rating	processor	CPU	Ram	Rar
0	0	0	HP	Victus 15-fb0157AX Gaming Laptop	49900	73.000000	5th Gen AMD Ryzen 5 5600H	Hexa Core, 12 Threads	8GB	
1	1	1	HP	15s-fq5007TU Laptop	39900	60.000000	12th Gen Intel Core i3 1215U	Hexa Core (2P + 4E), 8 Threads	8GB	
2	2	2	Acer	One 14 Z8-415 Laptop	26990	69.323529	11th Gen Intel Core i3 1115G4	Dual Core, 4 Threads	8GB	
3	3	3	Lenovo	Yoga Slim 6 14IAP8 82WU0095IN Laptop	59729	66.000000	12th Gen Intel Core i5 1240P	12 Cores (4P + 8E), 16 Threads	16GB	L
4	4	4	Apple	MacBook Air 2020 MGND3HN Laptop	69990	69.323529	Apple M1	Octa Core (4P + 4E)	8GB	

In [5]:
```
data.tail()
```

Out[5]:

	Unnamed: 0.1	Unnamed: 0	brand	name	price	spec_rating	processor	CPU	Ram	Ra
888	926	1015	Asus	Vivobook 15X 2023 K3504VAB-NJ321WS Laptop	44990	69.323529	13th Gen Intel Core i3 1315U	Hexa Core (2P + 4E), 8 Threads	8GB	
889	927	1016	Asus	TUF A15 FA577RM-HQ032WS Laptop	110000	71.000000	6th Gen AMD Ryzen 7 6800H	Octa Core, 16 Threads	16GB	
890	928	1017	Asus	ROG Zephyrus G14 2023 GA402XV-N2034WS Gaming L...	189990	89.000000	7th Gen AMD Ryzen 9 7940HS	Octa Core, 16 Threads	32GB	
891	929	1018	Asus	TUF Gaming F15 2023 FX507VU-LP083WS Gaming Laptop	129990	73.000000	13th Gen Intel Core i7 13700H	14 Cores (6P + 8E), 20 Threads	16GB	
892	930	1019	Asus	TUF Gaming A15 2023 FA577XU-LP041WS Gaming Laptop	131990	84.000000	7th Gen AMD Ryzen 9 7940HS	Octa Core, 16 Threads	16GB	

In [6]:

```
data.describe()
```

Out[6]:

	Unnamed: 0.1	Unnamed: 0	price	spec_rating	display_size	resolution_width	resoluti
count	893.000000	893.000000	893.000000	893.000000	893.000000	893.000000	8
mean	467.135498	521.382979	79907.409854	69.379026	15.173751	2035.393057	12
std	270.209769	299.916605	60880.043823	5.541555	0.939095	426.076009	3
min	0.000000	0.000000	9999.000000	60.000000	11.600000	1080.000000	7
25%	235.000000	265.000000	44500.000000	66.000000	14.000000	1920.000000	10
50%	467.000000	531.000000	61990.000000	69.323529	15.600000	1920.000000	10
75%	702.000000	784.000000	90990.000000	71.000000	15.600000	1920.000000	12
max	930.000000	1019.000000	450039.000000	89.000000	18.000000	3840.000000	34

In [7]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 893 entries, 0 to 892
Data columns (total 18 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Unnamed: 0.1       893 non-null    int64  
 1   Unnamed: 0         893 non-null    int64  
 2   brand              893 non-null    object 
 3   name               893 non-null    object 
 4   price              893 non-null    int64  
 5   spec_rating        893 non-null    float64
 6   processor          893 non-null    object 
 7   CPU                893 non-null    object 
 8   Ram                893 non-null    object 
 9   Ram_type           893 non-null    object 
 10  ROM                893 non-null    object 
 11  ROM_type           893 non-null    object 
 12  GPU                893 non-null    object 
 13  display_size       893 non-null    float64
 14  resolution_width   893 non-null    float64
 15  resolution_height  893 non-null    float64
 16  OS                 893 non-null    object 
 17  warranty           893 non-null    int64  
dtypes: float64(4), int64(4), object(10)
memory usage: 125.7+ KB
```

In [8]:
```
data.shape
```

Out[8]:
```
(893, 18)
```

In [9]:
```
list(data)
```

Out[9]:
```
['Unnamed: 0.1',
 'Unnamed: 0',
 'brand',
 'name',
 'price',
 'spec_rating',
 'processor',
 'CPU',
 'Ram',
 'Ram_type',
 'ROM',
 'ROM_type',
 'GPU',
 'display_size',
 'resolution_width',
 'resolution_height',
 'OS',
 'warranty']
```

In [10]:
```
data.isna().sum()
```

Out[10]:
```
Unnamed: 0.1         0
Unnamed: 0           0
brand                0
name                 0
price                0
spec_rating          0
processor            0
CPU                  0
Ram                  0
Ram_type             0
ROM                  0
ROM_type             0
GPU                  0
display_size         0
resolution_width     0
resolution_height    0
OS                   0
warranty             0
dtype: int64
```

In [11]:
```
data.head()
```

Out[11]:

	Unnamed: 0.1	Unnamed: 0	brand	name	price	spec_rating	processor	CPU	Ram	Rar
0	0	0	HP	Victus 15-fb0157AX Gaming Laptop	49900	73.000000	5th Gen AMD Ryzen 5 5600H	Hexa Core, 12 Threads	8GB	
1	1	1	HP	15s-fq5007TU Laptop	39900	60.000000	12th Gen Intel Core i3 1215U	Hexa Core (2P + 4E), 8 Threads	8GB	
2	2	2	Acer	One 14 Z8-415 Laptop	26990	69.323529	11th Gen Intel Core i3 1115G4	Dual Core, 4 Threads	8GB	
3	3	3	Lenovo	Yoga Slim 6 14IAP8 82WU0095IN Laptop	59729	66.000000	12th Gen Intel Core i5 1240P	12 Cores (4P + 8E), 16 Threads	16GB	L
4	4	4	Apple	MacBook Air 2020 MGND3HN Laptop	69990	69.323529	Apple M1	Octa Core (4P + 4E)	8GB	

In [12]:
```
data["brand"].unique()
```

Out[12]:
```
array(['HP', 'Acer', 'Lenovo', 'Apple', 'Dell', 'Asus', 'Samsung',
       'Ultimus', 'Primebook', 'MSI', 'Infinix', 'Wings', 'Honor',
       'Zebronics', 'Xiaomi', 'iBall', 'Chuwi', 'Realme', 'Avita',
       'Walker', 'Huawei', 'Tecno', 'Gigabyte', 'Vaio', 'Microsoft',
       'Fujitsu', 'LG', 'Ninkear', 'Razer', 'AXL'], dtype=object)
```

In [13]:
```
data.groupby(['brand']).count()
```

Out[13]:

	Unnamed: 0.1	Unnamed: 0	name	price	spec_rating	processor	CPU	Ram	Ram_type
brand									
AXL	2	2	2	2	2	2	2	2	2
Acer	84	84	84	84	84	84	84	84	84
Apple	16	16	16	16	16	16	16	16	16
Asus	157	157	157	157	157	157	157	157	157
Avita	1	1	1	1	1	1	1	1	1
Chuwi	3	3	3	3	3	3	3	3	3
Dell	107	107	107	107	107	107	107	107	107
Fujitsu	6	6	6	6	6	6	6	6	6
Gigabyte	8	8	8	8	8	8	8	8	8
HP	186	186	186	186	186	186	186	186	186
Honor	2	2	2	2	2	2	2	2	2
Huawei	2	2	2	2	2	2	2	2	2
Infinix	15	15	15	15	15	15	15	15	15
LG	9	9	9	9	9	9	9	9	9
Lenovo	169	169	169	169	169	169	169	169	169
MSI	65	65	65	65	65	65	65	65	65
Microsoft	2	2	2	2	2	2	2	2	2
Ninkear	1	1	1	1	1	1	1	1	1
Primebook	1	1	1	1	1	1	1	1	1
Razer	1	1	1	1	1	1	1	1	1
Realme	3	3	3	3	3	3	3	3	3
Samsung	28	28	28	28	28	28	28	28	28
Tecno	3	3	3	3	3	3	3	3	3
Ultimus	4	4	4	4	4	4	4	4	4
Vaio	1	1	1	1	1	1	1	1	1
Walker	1	1	1	1	1	1	1	1	1
Wings	3	3	3	3	3	3	3	3	3
Xiaomi	8	8	8	8	8	8	8	8	8
Zebronics	4	4	4	4	4	4	4	4	4
iBall	1	1	1	1	1	1	1	1	1

In [14]:

```
a=data.drop(['Unnamed: 0.1','Unnamed: 0','name','processor','CPU','Ram','Ram_type',
```

In [15]:

```
a
```

Out[15]:

	brand	price	spec_rating	ROM	ROM_type	display_size	resolution_width	resolution_hei
0	HP	49900	73.000000	512GB	SSD	15.6	1920.0	108
1	HP	39900	60.000000	512GB	SSD	15.6	1920.0	108
2	Acer	26990	69.323529	512GB	SSD	14.0	1920.0	108
3	Lenovo	59729	66.000000	512GB	SSD	14.0	2240.0	140
4	Apple	69990	69.323529	256GB	SSD	13.3	2560.0	160
...	...	...	...	...	...	...	...	
888	Asus	44990	69.323529	512GB	SSD	15.6	1920.0	108
889	Asus	110000	71.000000	1TB	SSD	15.6	2560.0	144
890	Asus	189990	89.000000	1TB	SSD	14.0	2560.0	160
891	Asus	129990	73.000000	512GB	SSD	15.6	1920.0	108
892	Asus	131990	84.000000	1TB	SSD	15.6	1920.0	108

893 rows × 9 columns

In [16]:
```
b=pd.get_dummies(a,dtype=int)
```

In [17]:
```
b
```

Out[17]:

	price	spec_rating	display_size	resolution_width	resolution_height	warranty	brand_AXL
0	49900	73.000000	15.6	1920.0	1080.0	1	0
1	39900	60.000000	15.6	1920.0	1080.0	1	0
2	26990	69.323529	14.0	1920.0	1080.0	1	0
3	59729	66.000000	14.0	2240.0	1400.0	1	0
4	69990	69.323529	13.3	2560.0	1600.0	1	0
...	...	...	...	...	...	...	...
888	44990	69.323529	15.6	1920.0	1080.0	1	0
889	110000	71.000000	15.6	2560.0	1440.0	1	0
890	189990	89.000000	14.0	2560.0	1600.0	1	0
891	129990	73.000000	15.6	1920.0	1080.0	1	0
892	131990	84.000000	15.6	1920.0	1080.0	1	0

893 rows × 45 columns

In [18]:
```
b.shape
```

Out[18]:
```
(893, 45)
```

In [19]:
```
y=b['price']
x=b.drop(['price'],axis=1)
```

In [20]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [21]:

```
x_train.head(100)
```

Out[21]:

	spec_rating	display_size	resolution_width	resolution_height	warranty	brand_AXL	brand_Ac
6	60.000000	15.6	1920.0	1080.0	1	0	
578	69.323529	15.6	1920.0	1080.0	1	0	
846	72.000000	14.0	2560.0	1600.0	1	0	
73	62.000000	15.6	1920.0	1080.0	1	0	
615	69.323529	15.6	1920.0	1080.0	1	0	
...	...	...	...	...	...	...	
177	75.000000	15.6	1920.0	1080.0	2	0	
649	76.000000	14.0	2880.0	1800.0	1	0	
711	65.000000	15.6	1920.0	1080.0	1	0	
616	76.000000	14.0	2880.0	1800.0	1	0	
383	60.000000	15.6	1920.0	1080.0	1	0	

100 rows × 44 columns

In [22]:

```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(x_train,y_train)
```

Out[22]:

```
▾ LinearRegression
LinearRegression()
```

In [23]:

```
ypred=reg.predict(x_test)
ypred
```

```
Out[23]: array([ 7.54001420e+04,  7.10766517e+04,  8.21909256e+04,  1.69615215e+05,
                 6.95951461e+04,  2.27160378e+04,  4.38228121e+04,  7.10766517e+04,
                 7.75249457e+04,  2.13166408e+03,  5.01049612e+04,  2.00689893e+05,
                 7.09811281e+04,  6.57720868e+04,  7.74318075e+04,  6.76079371e+04,
                 1.16456794e+05,  2.27160378e+04,  5.83771455e+04,  8.18651282e+03,
                 6.76079371e+04,  5.04384387e+04,  4.86139015e+04,  1.06798628e+05,
                 9.72806580e+04,  1.45268983e+05,  4.02190137e+04,  1.47298849e+05,
                 1.06384718e+05,  5.50297834e+04,  7.90516806e+04,  2.01152460e+05,
                 9.43766452e+04,  4.95629547e+04,  8.68028737e+04,  7.92533124e+04,
                 5.54381812e+04, -4.95641208e+03,  4.27386750e+04,  2.41629751e+05,
                 6.66575329e+04,  3.81393303e+04,  7.25327593e+04,  1.98515957e+05,
                 9.74702434e+04,  7.18834095e+04,  5.20921702e+04,  6.76079371e+04,
                 3.97457276e+04,  5.25005679e+04,  7.10766517e+04,  2.11582645e+05,
                 7.65544395e+04,  4.14163001e+04,  1.55793771e+05,  3.77389325e+04,
                 8.06962493e+04,  3.99090205e+04,  7.05656694e+04,  7.11293171e+04,
                 1.98920176e+04,  6.29299971e+04,  2.87167940e+04,  1.49592578e+05,
                 6.07823064e+04,  5.78446931e+04,  6.62395810e+04,  6.76079371e+04,
                 6.81917039e+04,  4.60942402e+04,  5.94612826e+04,  6.76079371e+04,
                 7.60014739e+04,  3.13228910e+04,  7.44622229e+04,  7.90516806e+04,
                 3.55410737e+04,  5.94612826e+04,  6.95693720e+04,  6.07823064e+04,
                 4.95629547e+04,  3.45920205e+04,  8.98517293e+04,  6.03900000e+04,
                 5.83771455e+04,  5.50407426e+04,  3.58003428e+05,  6.76079371e+04,
                 2.43286629e+05,  1.07643954e+05,  7.70045436e+04,  4.90318534e+04,
                 1.99600350e+04,  7.65544395e+04,  1.29750007e+05,  5.83771455e+04,
                 7.63156991e+04,  7.10766517e+04,  5.83771455e+04,  1.60802375e+05,
                 7.56096902e+04,  1.35031527e+05,  3.83485529e+04,  9.86645690e+04,
                 9.54734273e+04,  1.33982975e+05,  7.90516806e+04,  6.72391773e+04,
                 5.83771455e+04,  7.88480143e+04,  1.20727619e+05,  1.29750007e+05,
                 5.80995992e+04,  1.22416300e+05,  4.04646714e+04,  1.22078443e+05,
                 7.65544395e+04,  7.21903746e+04,  1.73721280e+05,  1.68483306e+05,
                 4.86139015e+04,  5.78446931e+04,  1.40991294e+05,  7.10766517e+04,
                 6.76079371e+04,  1.45693832e+04,  7.38133525e+04,  4.91655161e+04,
                 1.31572124e+05,  6.04755969e+04,  5.83771455e+04,  3.91342865e+04,
                 1.75069965e+04,  3.55410737e+04,  1.29394986e+05,  6.76079371e+04,
                 7.67509658e+04,  1.35813609e+05,  5.06866057e+04,  8.38652604e+04,
                 5.02304910e+04,  1.28634511e+05,  3.68634486e+04,  4.96134978e+04,
                 5.01049612e+04,  3.74041040e+04,  1.48011964e+05,  2.27160378e+04,
                 1.22872676e+05,  1.15566318e+05,  1.53406698e+05,  4.22917841e+04,
                 8.14849167e+04,  3.69276132e+04,  4.57404530e+04,  1.17790006e+05,
                 5.64412509e+04,  3.44664907e+04,  2.12487124e+04,  1.35207123e+05,
                 1.16147962e+05,  5.61427694e+04,  6.15683138e+04,  1.89412164e+05,
                 2.04446097e+04,  7.30638606e+04,  1.01025852e+05,  2.13200937e+04,
                 6.72391773e+04,  1.74713941e+05,  4.93913291e+04,  4.72915266e+04,
                 5.32051561e+04,  7.85473034e+04,  1.66296242e+04,  6.07823064e+04,
                 5.25005679e+04,  3.91342865e+04,  6.76079371e+04,  6.42510209e+04,
                 5.04384387e+04,  7.09050261e+04,  6.22818814e+04,  8.62446472e+04,
                 5.94612826e+04,  2.39539852e+04,  4.72809410e+04,  7.90516806e+04,
                 2.81147735e+04,  4.43539134e+04,  1.54998469e+04,  5.79398576e+03,
                 6.66575329e+04,  7.30638606e+04,  1.07591289e+05,  1.12575449e+05,
                 6.75087399e+04,  2.38284555e+04,  4.24611287e+04,  1.94921826e+05,
                 7.10766517e+04,  3.26581998e+02,  5.02304910e+04,  6.62395810e+04,
                 1.30141791e+05,  7.40669303e+04,  2.11275327e+05,  1.35532308e+05,
                 6.76079371e+04,  2.21399998e+05,  1.13725358e+05,  5.50803796e+04,
                 4.35161026e+04,  7.90516806e+04,  7.85473034e+04,  3.15288775e+04,
                 7.90516806e+04,  4.91545570e+04,  5.83771455e+04,  5.40249450e+04,
                 3.31847768e+04,  5.01049612e+04,  6.07823064e+04,  6.66575329e+04,
                 6.13134077e+04,  6.76079371e+04,  5.50710195e+04,  6.76079371e+04,
                 1.56818009e+04,  4.19583067e+04,  6.42510209e+04,  6.76079371e+04,
                 5.04384387e+04,  5.50803796e+04,  1.76210326e+05,  9.53260011e+04,
                 2.09028241e+05,  1.64157941e+05,  8.36338625e+04,  7.60520170e+04,
                 3.55410737e+04,  5.89587899e+04,  5.02304910e+04,  5.01049612e+04,
                 1.42238496e+05,  4.32793305e+04,  2.38284555e+04,  1.59435702e+05,
                 5.02668130e+04,  1.01602182e+05,  7.51279879e+04,  1.15566318e+05,
                 9.50481749e+04,  6.07823064e+04,  7.70045436e+04,  4.93913291e+04,
                 6.76079371e+04,  6.13134077e+04,  5.97968160e+04,  8.13919915e+04,
                 2.38284555e+04,  3.44664907e+04,  2.89542866e+05,  1.04138841e+05,
                 2.89247417e+04,  3.68634486e+04,  3.52017170e+04,  1.75677910e+05,
                 8.29910125e+04,  2.74400993e+03,  1.14059836e+05,  4.03417172e+04,
                 7.96333250e+04,  2.13960796e+05,  6.24277198e+04,  8.39765028e+04,
                 7.90516806e+04,  2.89868922e+05,  7.10766517e+04,  3.79475856e+04,
                 7.70045436e+04,  3.51326759e+04,  7.78265817e+04,  5.78648122e+04,
                 6.70170085e+04,  1.54927149e+05,  5.50297834e+04,  7.65544395e+04,
                 1.17357503e+05,  3.74041040e+04,  7.50159835e+04,  4.98911344e+04,
                 5.23289423e+04,  2.10044353e+04,  4.36877282e+04])
```

In [24]:

```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[24]:

```
0.687177984821383
```

In [25]:

```
from sklearn.metrics import mean_squared_error
l=mean_squared_error(ypred,y_test)
```

In [26]:

```
l
```

Out[26]:

```
1233513475.5441575
```

In [27]:

```
Results=pd.DataFrame(columns=['price','predicted'])
Results['price']=y_test
Results['predicted']=ypred
#Results['km']=x_test['km']
Results=Results.reset_index()
Results['ID']=Results.index
Results.head(15)
```

Out[27]:

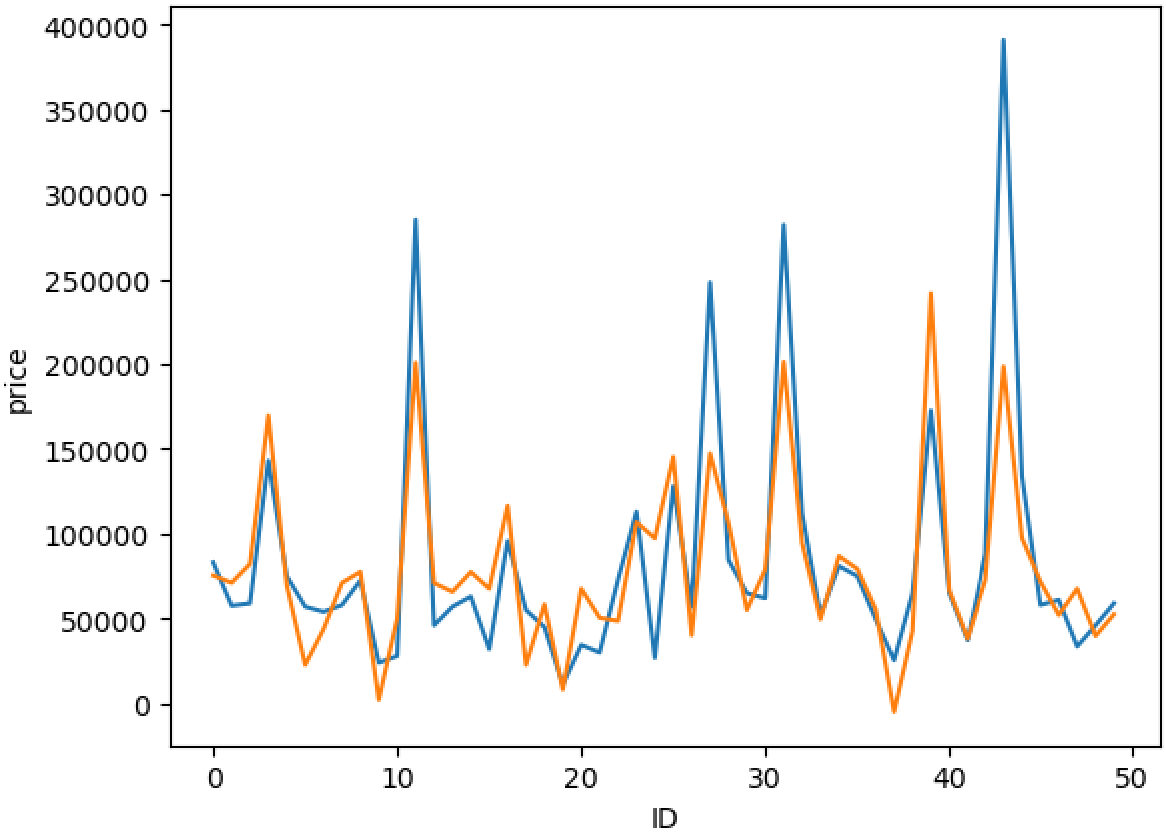
	index	price	predicted	ID
0	710	83090	75400.142024	0
1	440	57580	71076.651679	1
2	525	58990	82190.925606	2
3	721	142990	169615.215143	3
4	39	74990	69595.146102	4
5	290	56990	22716.037779	5
6	300	53990	43822.812098	6
7	333	57990	71076.651679	7
8	208	72490	77524.945702	8
9	136	23990	2131.664076	9
10	137	27990	50104.961242	10
11	697	284990	200689.892640	11
12	486	45999	70981.128067	12
13	244	56990	65772.086758	13
14	344	62990	77431.807467	14

In [28]:

```
import seaborn as sb
```

```
In [29]: import matplotlib.pyplot as plt
         sb.lineplot(x='ID',y='price',data=Results.head(50))
         sb.lineplot(x='ID',y='predicted',data=Results.head(50))
         plt.plot()
```

```
Out[29]: []
```



```
In [30]: cor=b.corr()
         cor
```

Out[30]:

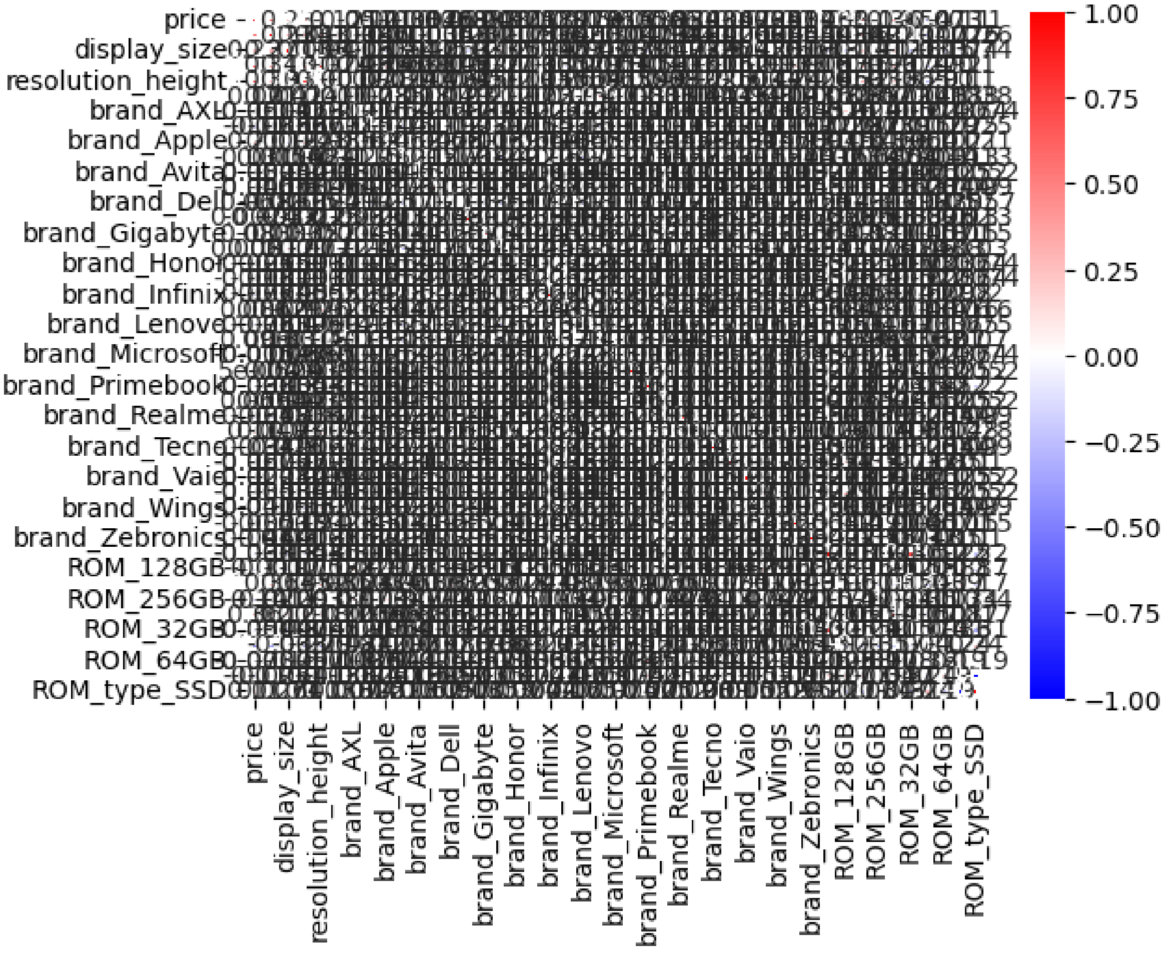
	price	spec_rating	display_size	resolution_width	resolution_height	warrant
price	1.000000	0.546391	0.233815	0.586042	0.604748	0.11710
spec_rating	0.546391	1.000000	0.274206	0.337649	0.328525	0.10950
display_size	0.233815	0.274206	1.000000	0.125088	0.029692	0.04112
resolution_width	0.586042	0.337649	0.125088	1.000000	0.731557	0.02419
resolution_height	0.604748	0.328525	0.029692	0.731557	1.000000	-0.00106
warranty	0.117101	0.109501	0.041126	0.024199	-0.001060	1.00000
brand_AXL	-0.050938	-0.000475	-0.016343	-0.012838	-0.020068	-0.01152
brand_Acer	-0.112569	-0.035680	-0.033495	-0.066400	-0.070893	-0.07840
brand_Apple	0.209386	-0.010454	-0.119446	0.242524	0.278321	-0.08453
brand_Asus	-0.031374	0.011010	0.053663	0.067806	0.024313	-0.11237
brand_Avita	-0.031871	-0.000336	-0.041872	-0.009073	-0.014182	-0.00814
brand_Chuwi	-0.046037	-0.000582	-0.049925	-0.000463	0.007646	-0.01412
brand_Dell	0.068179	-0.019738	-0.062572	0.015085	-0.023829	-0.07922
brand_Fujitsu	0.002364	-0.003154	-0.123301	-0.022287	-0.014689	0.23168
brand_Gigabyte	0.083397	0.105076	0.033049	0.072475	0.051436	0.19508
brand_HP	0.008135	0.001655	0.123348	-0.069829	-0.122679	-0.11635
brand_Honor	-0.025242	-0.014690	-0.059250	-0.012838	-0.011363	-0.01152
brand_Huawei	-0.017838	-0.037460	-0.018867	-0.012838	-0.011363	-0.01152
brand_Infinix	-0.077156	-0.064150	-0.044622	-0.035419	-0.055363	-0.03180
brand_LG	0.080778	0.061919	0.029101	-0.117908	0.255245	-0.02455
brand_Lenovo	-0.078185	-0.062592	-0.040401	-0.029213	-0.068779	0.01367
brand_MSI	0.096435	0.132546	0.133215	0.001816	-0.041064	0.49904
brand_Microsoft	-0.015158	-0.000475	-0.008771	0.051246	0.064657	-0.01152
brand_Ninkear	0.000050	0.040027	0.029476	0.041248	0.039132	-0.00814
brand_Primebook	-0.037924	-0.000336	-0.127490	-0.052633	-0.046170	-0.00814
brand_Razer	0.066079	0.094436	-0.041872	0.041248	0.039132	-0.00814
brand_Realme	-0.033948	-0.048268	-0.072607	-0.015733	0.082077	-0.01412
brand_Samsung	0.111194	0.041363	-0.028515	0.030918	0.117403	-0.04377
brand_Tecno	-0.034263	-0.042410	0.026367	-0.015733	-0.024591	-0.01412
brand_Ultimus	-0.067019	-0.000672	-0.076739	-0.061810	-0.060454	-0.01632
brand_Vaio	-0.021966	-0.000336	-0.041872	-0.009073	-0.014182	0.09431
brand_Walker	-0.034622	-0.000336	-0.038305	-0.009073	-0.014182	-0.00814
brand_Wings	-0.040633	-0.000582	-0.006624	-0.130258	0.124744	-0.01412
brand_Xiaomi	-0.036507	-0.053021	-0.017601	0.186342	0.244166	-0.02313
brand_Zebronics	-0.046211	-0.043952	0.030463	-0.018177	-0.028412	-0.01632

	price	spec_rating	display_size	resolution_width	resolution_height	warrant
brand_iBall	-0.038469	-0.000336	-0.127490	-0.052633	-0.046170	-0.00814
ROM_128GB	-0.102276	-0.001169	-0.104504	-0.070543	-0.053484	-0.02839
ROM_1TB	0.482659	0.364440	0.141455	0.345545	0.317134	0.08450
ROM_256GB	-0.138886	-0.021917	-0.113826	-0.092967	-0.033457	-0.07024
ROM_2TB	0.355126	0.202100	0.120896	0.207873	0.179120	0.07180
ROM_32GB	-0.054049	-0.000475	-0.180399	-0.074476	-0.065331	-0.01152
ROM_512GB	-0.407571	-0.363590	-0.032862	-0.274487	-0.288811	-0.04839
ROM_64GB	-0.073170	-0.000752	-0.145007	-0.092017	-0.010093	-0.01825
ROM_type_Hard-Disk	-0.105690	-0.025886	-0.073607	-0.101228	-0.102298	-0.03775
ROM_type_SSD	0.105690	0.025886	0.073607	0.101228	0.102298	0.03775

45 rows × 45 columns

In [31]:
```
import seaborn as sb
sb.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidth=5,cmap="bwr")
```

Out[31]:
```
<Axes: >
```



In [ ]: