In [1]:
```
import pandas as pd
import numpy as np
import warnings
import pickle
warnings.filterwarnings('ignore')
```

In [2]:
```
data=pd.read_csv("C:\\Users\\akhil\\Downloads\\laptop.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.describe()
```

Out[4]:

	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 [5]:
```
data.head()
```

Out[5]:

	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 [6]:
```
data.tail()
```

Out[6]:

	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 [7]:

```
data.shape
```

Out[7]:

```
(893, 18)
```

In [8]:

```
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 [9]:
```
data['price'].unique()
```

```
Out[9]: array([ 49900,  39900,  26990,  59729,  69990,  39990,  36790,  76990,
                48990,  74990,  49990,  10990,  72990,  29990,  53649,  33990,
                79990,  68990,  45990, 104990,  54990,  49580,  55990,  43990,
                19850,  28990,  54999,  70990,  56889, 179880,  53790,  37990,
                96990,  71990,  59990,  62990,  30990,  36490,  77990,  41190,
                89990, 399999,  31990,  31490,  70590,  23990,  37999,  52990,
                30849,  58990,  34990,  51780,  85990, 148990,  84490,  60990,
                35990,  65990,  61990,  36990,  50970, 184999, 109990,  42990,
                39150,  92999,  80990,  55850,  38990,  53990, 139234,  57990,
                89999,  92990,  59089,  72999, 126742, 126990,  74390,  98174,
                18990,  41950,  46990,   9999,  40990,  88990,  80739,  69501,
                64990,  37729,  50712,  44990,  52101,  69490,  39965,  46999,
                83990,  51499,  50990,  27990,  84390,  24990, 281990,  26590,
                71189,  85999,  93990,  84990,  44190, 159990,  73990,  57500,
                75990, 144990,  60539,  56990,  75533,  94990, 179990, 129999,
                47999, 114499,  66490,  86490,  32790, 429990, 102990,  77889,
               138990,  51990,  50400, 108000,  47990,  91990,  25449,  33207,
                42434,  42200,  41990,  80900, 135990,  72490,  45989,  52500,
                59999, 112990,  21990, 201498,  68999, 107690,  25490,  38999,
                55500, 124999, 136990, 131990, 188752,  36999,  92890,  51999,
                76890,  37790,  17990,  86189, 194990,  51390,  76900,  27890,
                24999,  33999,  31890,  62390,  43490,  95640, 125990,  99999,
                65999,  37980,  40789, 450039,  16990,  67000,  34499, 107763,
                61490,  80490,  44500,  48999, 116990,  33790,  34099,  65600,
               161590,  98990, 172990,  25990, 126790, 209990, 142990, 156990,
                93980,  46490,  90490,  99990,  80500,  34250,  58999, 420000,
                42999,  38861,  33590,  15990,  33980,  78400,  45889,  90990,
               133990, 165990, 163990, 139990, 149990, 155990, 229900,  34999,
               290990, 234990,  82799,  31790,  86990,  52558,  94490,  33800,
               104490, 105739,  45500,  50760,  33199,  43999,  32990, 179490,
                83249,  85490,  55890,  40689,  66648, 118499,  35490,  62889,
                63999, 119990,  85590,  55800,  72422,  31500,  63490,  57580,
               339990,  35999,  41710,  50090,  45890,  37672,  83523,  52128,
                54790, 147090,  67990,  49290,  83149,  66990,  51573,  29000,
               239900, 309490,  40250,  37290,  60390,  90464,  54490, 199900,
               260000,  45999, 106990, 221990, 146990,  64490,  78990,  59900,
                19990, 101990,  33499, 215994,  26500,  68000,  47925,  67800,
                60490, 109999,  87990, 121900, 136490, 143700,  50490, 136899,
               154990,  50000,  76500, 103221,  44980,  26920,  14490, 362999,
               169999, 189999, 298999, 267999, 108990,  62608,  57999, 297990,
               138490, 103395, 419990,  13990,  95689,  26732,  65391,  90000,
                38390, 129990,  49989, 259990,  44690, 123300,  74499, 127990,
                91590, 103060,  39600, 137590,  53980,  28099, 415000,  52999,
                88999, 111490,  89500,  66388,  36480,  63989,  64800,  14990,
                22990,  42900, 130000, 104222,  63990,  47100,  43800, 164990,
               304990,  87490, 107990,  46900,  98890,  58490,  69199,  97990,
               116463, 153625, 270129, 107190,  32999, 102090, 231746,  81990,
                36890, 117574, 284990, 344990, 146599, 128599, 151990, 119888,
                76888, 256990,  83090,  67999,  96256, 323290,  82990, 199990,
                84800, 114990,  80000,  98999,  29999,  97705, 240707, 390914,
               172588, 254657, 152499,  78299, 115990,  26999, 103099, 130800,
                38500,  77490,  44000,  57799, 171990, 163999,  43890,  79999,
               137890,  83500, 111990,  46200,  52998,  45499,  49999,  29980,
                57290,  47890,  95490, 159000,  80610, 303490, 201990, 278290,
               286200, 157544, 140990, 133000,  72500, 127000, 205000,  89799,
                12990,  41490, 208990,  91560,  74000,  32490,  74888,  66500,
               156799, 166899, 171499,  35998,  44989,  71570,  64590, 247999,
                34390,  28490, 107399, 129199, 187490, 125699, 110000, 189990],
              dtype=int64)
```

```
In [10]: data['brand'].unique()
```

Out[10]:

```
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 [11]:

```python
data.groupby(['price']).count()
```

Out[11]:

	Unnamed: 0.1	Unnamed: 0	brand	name	spec_rating	processor	CPU	Ram	Ram_type	ROM
price										
9999	1	1	1	1	1	1	1	1	1	1
10990	3	3	3	3	3	3	3	3	3	3
12990	1	1	1	1	1	1	1	1	1	1
13990	1	1	1	1	1	1	1	1	1	1
14490	1	1	1	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...
415000	1	1	1	1	1	1	1	1	1	1
419990	1	1	1	1	1	1	1	1	1	1
420000	1	1	1	1	1	1	1	1	1	1
429990	1	1	1	1	1	1	1	1	1	1
450039	1	1	1	1	1	1	1	1	1	1

464 rows × 17 columns

In [12]:

```python
data1=data.drop(['resolution_width','resolution_height','display_size'],axis=1)
data1
```

Out[12]:

	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 × 15 columns

In [13]:
```
data1.shape
```
Out[13]:
```
(893, 15)
```

In [14]:
```
data2=pd.get_dummies(data1,dtype=int)
data2
```
Out[14]:

	Unnamed: 0.1	Unnamed: 0	price	spec_rating	warranty	brand_AXL	brand_Acer	brand_Apple
0	0	0	49900	73.000000	1	0	0	0
1	1	1	39900	60.000000	1	0	0	0
2	2	2	26990	69.323529	1	0	1	0
3	3	3	59729	66.000000	1	0	0	0
4	4	4	69990	69.323529	1	0	0	1
...	...	...	...	...	...	...	...	...
888	926	1015	44990	69.323529	1	0	0	0
889	927	1016	110000	71.000000	1	0	0	0
890	928	1017	189990	89.000000	1	0	0	0
891	929	1018	129990	73.000000	1	0	0	0
892	930	1019	131990	84.000000	1	0	0	0

893 rows × 1239 columns

In [15]:
```
data2.shape
```
Out[15]:
```
(893, 1239)
```

In [16]:
```
y=data2['price']
x=data2.drop('price',axis=1)
```

In [17]:
```
y
```
Out[17]:
```
0       49900
1       39900
2       26990
3       59729
4       69990
        ...  
888     44990
889    110000
890    189990
891    129990
892    131990
Name: price, Length: 893, dtype: int64
```

In [18]:
```
x
```

Out[18]:

	Unnamed: 0.1	Unnamed: 0	spec_rating	warranty	brand_AXL	brand_Acer	brand_Apple	brand_A
0	0	0	73.000000	1	0	0	0	
1	1	1	60.000000	1	0	0	0	
2	2	2	69.323529	1	0	1	0	
3	3	3	66.000000	1	0	0	0	
4	4	4	69.323529	1	0	0	1	
...	...	...	...	...	...	...	...	
888	926	1015	69.323529	1	0	0	0	
889	927	1016	71.000000	1	0	0	0	
890	928	1017	89.000000	1	0	0	0	
891	929	1018	73.000000	1	0	0	0	
892	930	1019	84.000000	1	0	0	0	

893 rows × 1238 columns

In [19]:
```
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 [20]:
```
x_train.head()
```

Out[20]:

	Unnamed: 0.1	Unnamed: 0	spec_rating	warranty	brand_AXL	brand_Acer	brand_Apple	brand_A
6	6	6	60.000000	1	0	0	0	
578	603	683	69.323529	1	0	0	0	
846	884	972	72.000000	1	0	0	0	
73	75	78	62.000000	1	0	0	0	
615	644	726	69.323529	1	0	0	0	

5 rows × 1238 columns

In [21]:
```
y_train.head()
```

Out[21]:
```
6       36790
578     38390
846    208990
73      36990
615     35990
Name: price, dtype: int64
```

In [22]:
```
x_test.head()
```

Out[22]:

	Unnamed: 0.1	Unnamed: 0	spec_rating	warranty	brand_AXL	brand_Acer	brand_Apple	brand_A
710	744	828	60.000000	1	0	0	0	
440	461	525	69.323529	1	0	0	0	
525	549	627	65.000000	1	0	0	0	
721	756	841	82.000000	1	0	0	0	
39	40	41	70.000000	1	0	0	0	

5 rows × 1238 columns

In [23]:

```
y_test.head()
```

Out[23]:

```
710      83090
440      57580
525      58990
721     142990
39       74990
Name: price, dtype: int64
```

In [24]:

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

Out[24]:

```
▾ LinearRegression
LinearRegression()
```

In [25]:

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

In [26]:

```
ypred
```

```
Out[26]: array([ 84779.91826773,  84892.96603538,  94825.28272375, 130256.91403921,
                 72099.69118276,  60256.36680238,  21648.54933408,  51506.24486265,
                 58778.74762939,  15970.27345092,  42261.09513357, 372695.13951805,
                 77069.99578723,  50572.99178714,  79531.09196174,  24384.14064905,
                120237.07809639,  60740.63352907,  46064.1124315 ,   2985.96587669,
                 37689.93893687,  28910.97559983,  58432.31222861, 121330.57097667,
                 14387.47086702, 126837.90683785,  52332.27649558, 190899.49120503,
                132626.8078261 ,  77923.84173443,  59086.12153453, 232027.49198722,
                105871.75087911,  54137.30060173, 108825.49901407,  73593.19957579,
                 68137.41663335,  12856.58737942,  63412.66106665, 198517.12455949,
                 74259.30212369,  44661.10856676,  94007.33977539, 230684.78560198,
                125769.89559335,  69139.01521391,  64771.85017651,  42758.03395069,
                 31470.62913337,  51741.75889547,  47325.97283899, 178551.66768419,
                 89972.2825418 ,  41531.20845029, 103347.58351658,  47395.52922817,
                 71481.8747592 ,  44712.05667281,  72928.3574883 ,  87294.40484286,
                 46774.37053975,  65360.31280359,  39568.63628699, 157220.88951633,
                 60082.07100878,  62053.96155015,  63823.16983109,  43327.10150526,
                 86053.5660023 ,  46185.86017827,  54395.21861136,  38376.9065651 ,
                 65270.94898709,  18002.06385217,  72084.81955055,  43226.96276895,
                 45567.74961334,  54599.79975962,  39605.78187749,  42977.9693687 ,
                 52909.5732628 ,  43528.24366506, 118341.0584762 ,  60155.18159287,
                 44097.90223846,  64198.70955   , 191036.54737483,  40528.98577014,
                252595.05511944, 104701.31797034,  71232.57228955,  38422.1450783 ,
                 15929.13711818,  89843.52162146, 138247.62730437,  39415.4602542 ,
                 87961.08092766,  34442.66998219,  52407.53844864, 300277.48199531,
                 59854.30456686, 138684.77325738,  18109.68271211, 123038.12294654,
                 76692.10680265,  82051.46240879,  44384.79971689,  74750.99308319,
                 38698.79498419,  16378.25750707, 127527.86350983, 137595.18149915,
                 55183.5554836 , 100991.63147322,  68660.00861292,  93675.55420295,
                 50866.9476701 ,  35886.04701054, 169032.62919775, 158363.36029415,
                 56721.24438845,  62885.4874842 , 138506.92244261,  45900.73442455,
                 58453.70038846,  41060.13117523,  62683.87925445,  94980.0552122 ,
                124350.0324077 ,  30019.26081896,  42948.29616571,  56073.15433466,
                 36686.02772009,  45665.16589657,  84214.56267327,  43035.36318773,
                 74519.29107868, 127931.2167314 ,  24867.41681663, 105011.78902279,
                 36326.77957148, 118815.21430171,  70608.99935592,  51455.05901839,
                 65084.81864189,  59262.6104155 , 103113.12545163,  39089.38136409,
                210367.42061086,  71076.44571573, 192986.03708939,  29710.29199406,
                106100.23379634,  51934.05871047,  69374.14132821, 105605.06647363,
                 59956.65781541,  40190.09038379,  34785.18006993, 110733.44826575,
                 76159.0634429 ,  56137.9601469 ,  74796.32003788, 235421.32550155,
                 42974.32408102,  58223.93175727,  90484.61684947,  38923.18574411,
                 56649.41570537, 152312.99033392,  68389.80576472,  63195.68011034,
                 53733.76407603,  52875.4402738 ,  67291.20297072,  50495.43362791,
                 47468.17297919,  46205.29737771,  78283.66952451,  58101.730276  ,
                 55378.48830583,  38026.8467612 ,  56449.87001966,  86239.79227284,
                 84715.95023146,  34290.17142803,  20303.87912696,  84141.00686442,
                 45325.46732468,  47453.9099735 ,  27978.85029121,   8764.72274451,
                 54136.97919313,  72941.12016929,  91188.81365331, 127972.02324119,
                 68309.59192956,  22205.25927206,  70426.85150674, 259564.04003016,
                 57735.95041513,  36898.10335843,  54366.8755634 ,  77401.49143197,
                163581.55082732,  82802.46745757, 293792.40215525, 109881.82276112,
                 55781.13617041,  84229.25692761,  84480.60357556, 104758.55602447,
                 54173.65481268,  78227.77140525,  50760.16321025,  76449.74919973,
                 37811.75060045,  53995.76553022,  38690.50244895,  31997.27190358,
                 23722.16827035,  34558.75062597,  48637.84060003,  80672.02921733,
                 81443.28710729,  38914.54201722,  25099.78373576,  43568.68937922,
                 41390.29574401,  45244.92815588,  63998.21217784,  57814.37808943,
                 27134.97975639,  84694.2354708 , 129970.57329726,  79317.84885356,
                147244.98249741, 143834.29640324,  80774.00884725,  66918.58885611,
                 44626.72678023,  58733.30323263,  41551.24710361,  47885.37491441,
                 93845.85023072,  75270.41716399,  27169.16260216, 192432.75542341,
                 68432.80035355,  85158.57596538,  65870.58629663,  45099.81040606,
                101008.96156776, 119794.41616012,  89918.4097638 ,  53434.89951222,
                 41031.47288344,  82183.55257902,  73871.97452452,  86762.68449562,
                 38173.2032002 ,  68035.48755453, 416430.49299227, 100287.01373193,
                 48070.93666591,  52968.31589373,  32086.87004482, 120047.82542749,
                 49637.80489251,  32098.51992136, 153288.61089369,  36212.32687105,
                 88555.37429933, 175554.71475373,  56338.12212957,  90209.87460746,
                 49953.80497897, 198526.39683162,  71646.82633941,  44818.42670131,
                 69827.69346036,  48287.72225818, 122934.60540116,  37142.4318674 ,
                 50252.94286728, 281119.00884849,  57975.27228406,  67658.28643344,
                 58796.41721038,  36358.06422445,  94710.79751734,  57431.19467487,
                 48651.25357351,  16417.25329807,  33585.10646047])
```

In [27]:
```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[27]:
```
0.7975905080369208
```

In [28]:
```
from sklearn.metrics import mean_squared_error
l=mean_squared_error(ypred,y_test)
```

In [29]:
```
l
```

Out[29]:
```
798137035.7579992
```

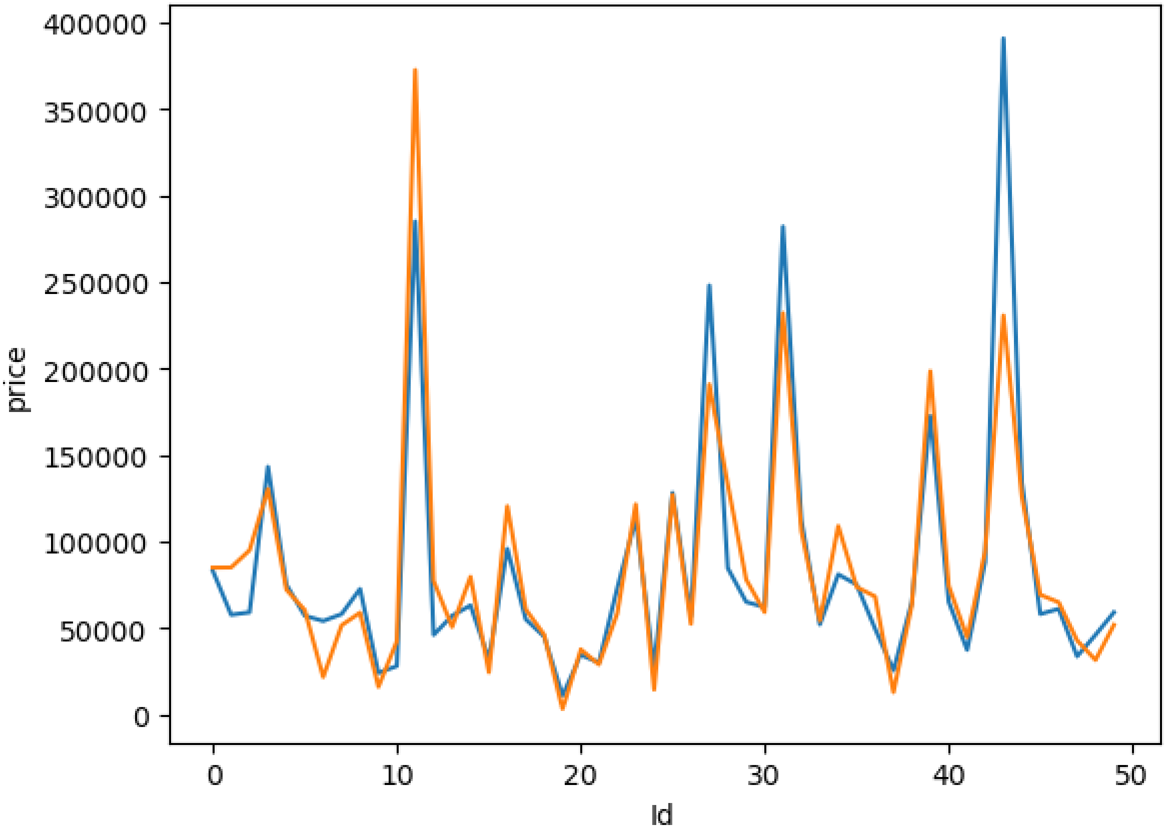
In [30]:
```
#Results= pd.DataFrame(columns=['Actual','predicted'])
#Results['Actual']=y_test
Results=pd.DataFrame(columns=['price','predicted'])
Results['price']=y_test
Results['predicted']=ypred
#Results['Km']=x_test['Km']
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(15)
```

Out[30]:

	index	price	predicted	Id
0	710	83090	84779.918268	0
1	440	57580	84892.966035	1
2	525	58990	94825.282724	2
3	721	142990	130256.914039	3
4	39	74990	72099.691183	4
5	290	56990	60256.366802	5
6	300	53990	21648.549334	6
7	333	57990	51506.244863	7
8	208	72490	58778.747629	8
9	136	23990	15970.273451	9
10	137	27990	42261.095134	10
11	697	284990	372695.139518	11
12	486	45999	77069.995787	12
13	244	56990	50572.991787	13
14	344	62990	79531.091962	14

In [31]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='Id',y='price',data=Results.head(50))
sns.lineplot(x='Id',y='predicted',data=Results.head(50))
plt.plot()
```

Out[31]:
```
[]
```



In [32]:
```
cor=data2.corr()
cor
```

Out[32]:

	Unnamed: 0.1	Unnamed: 0	price	spec_rating	warranty	brand_AXL	brand_Acer	br
Unnamed: 0.1	1.000000	0.999665	0.162473	0.109777	0.157482	0.071466	-0.049608	
Unnamed: 0	0.999665	1.000000	0.162619	0.108691	0.158614	0.069722	-0.049247	
price	0.162473	0.162619	1.000000	0.546391	0.117101	-0.050938	-0.112569	
spec_rating	0.109777	0.108691	0.546391	1.000000	0.109501	-0.000475	-0.035680	
warranty	0.157482	0.158614	0.117101	0.109501	1.000000	-0.011528	-0.078402	
...	...	...	...	...	...	...	...	
OS_Windows 10 OS	0.012044	0.013284	-0.029673	-0.077755	-0.025893	-0.005042	-0.034291	
OS_Windows 10 OS	0.012544	0.016962	-0.034228	-0.045344	0.054543	-0.008524	-0.035963	
OS_Windows 11 OS	0.094991	0.095620	0.041161	0.018773	0.128195	-0.006193	-0.042118	
OS_Windows 11 OS	-0.058015	-0.062330	0.016752	0.073387	-0.012201	0.017850	0.051639	
OS_Windows OS	0.021454	0.023397	-0.026647	-0.034709	-0.024550	-0.004780	0.044297	

1239 rows × 1239 columns

In [ ]:

```
import seaborn as sns
sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidth=10,cmap='bwr')
```

In [ ]: