In [1]:
```
import pandas as pd
```

In [2]:
```
data = pd.read_csv('mobile phone price prediction.csv')
```

In [3]:
```
data
```

Out[3]:

	Unnamed: 0	Name	Rating	Spec_score	No_of_sim	Ram	Battery	Display	Camera	E
0	0	Samsung Galaxy F14 5G	4.65	68	Dual Sim, 3G, 4G, 5G, VoLTE,	4 GB RAM	6000 mAh Battery	6.6 inches	50 MP + 2 MP Dual Rear & 13 MP Front Camera	S
1	1	Samsung Galaxy A11	4.20	63	Dual Sim, 3G, 4G, VoLTE,	2 GB RAM	4000 mAh Battery	6.4 inches	13 MP + 5 MP + 2 MP Triple Rear & 8 MP Fro...	
2	2	Samsung Galaxy A13	4.30	75	Dual Sim, 3G, 4G, VoLTE,	4 GB RAM	5000 mAh Battery	6.6 inches	50 MP Quad Rear & 8 MP Front Camera	S
3	3	Samsung Galaxy F23	4.10	73	Dual Sim, 3G, 4G, VoLTE,	4 GB RAM	6000 mAh Battery	6.4 inches	48 MP Quad Rear & 13 MP Front Camera	S
4	4	Samsung Galaxy A03s (4GB RAM + 64GB)	4.10	69	Dual Sim, 3G, 4G, VoLTE,	4 GB RAM	5000 mAh Battery	6.5 inches	13 MP + 2 MP + 2 MP Triple Rear & 5 MP Fro...	S
...	...	...	...	...	...	...	...	...	...	
1365	1365	TCL 40R	4.05	75	Dual Sim, 3G, 4G, 5G, VoLTE,	4 GB RAM	5000 mAh Battery	6.6 inches	50 MP + 2 MP + 2 MP Triple Rear & 8 MP Fro...	
1366	1366	TCL 50 XL NxtPaper 5G	4.10	80	Dual Sim, 3G, 4G, VoLTE,	8 GB RAM	5000 mAh Battery	6.8 inches	50 MP + 2 MP Dual Rear & 16 MP Front Camera	

	Unnamed: 0	Name	Rating	Spec_score	No_of_sim	Ram	Battery	Display	Camera	E
1367	1367	TCL 50 XE NxtPaper 5G	4.00	80	Dual Sim, 3G, 4G, 5G, VoLTE,	6 GB RAM	5000 mAh Battery	6.6 inches	50 MP + 2 MP Dual Rear & 16 MP Front Camera	S
1368	1368	TCL 40 NxtPaper 5G	4.50	79	Dual Sim, 3G, 4G, 5G, VoLTE,	6 GB RAM	5000 mAh Battery	6.6 inches	50 MP + 2 MP + 2 MP Triple Rear & 8 MP Fro...	S
1369	1369	TCL Trifold	4.65	93	Dual Sim, 3G, 4G, 5G, VoLTE, Vo5G,	12 GB RAM	4600 mAh Battery	10 inches	Foldable Display, Dual Display	

1370 rows × 18 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1370 entries, 0 to 1369
Data columns (total 18 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Unnamed: 0         1370 non-null   int64  
 1   Name               1370 non-null   object 
 2   Rating             1370 non-null   float64
 3   Spec_score         1370 non-null   int64  
 4   No_of_sim          1370 non-null   object 
 5   Ram                1370 non-null   object 
 6   Battery            1370 non-null   object 
 7   Display            1370 non-null   object 
 8   Camera             1370 non-null   object 
 9   External_Memory    1370 non-null   object 
 10  Android_version    927 non-null    object 
 11  Price              1370 non-null   object 
 12  company            1370 non-null   object 
 13  Inbuilt_memory     1351 non-null   object 
 14  fast_charging      1281 non-null   object 
 15  Screen_resolution  1368 non-null   object 
 16  Processor          1342 non-null   object 
 17  Processor_name     1370 non-null   object 
dtypes: float64(1), int64(2), object(15)
memory usage: 192.8+ KB
```

In [7]:
```
data.drop('Unnamed: 0',axis =1,inplace = True)
```

In [8]:
```
data
```

Out[8]:

	Name	Rating	Spec_score	No_of_sim	Ram	Battery	Display	Camera	External_Mem
0	Samsung Galaxy F14 5G	4.65	68	Dual Sim, 3G, 4G, 5G, VoLTE,	4 GB RAM	6000 mAh Battery	6.6 inches	50 MP + 2 MP Dual Rear & 13 MP Front Camera	Memory C Supported, up
1	Samsung Galaxy A11	4.20	63	Dual Sim, 3G, 4G, VoLTE,	2 GB RAM	4000 mAh Battery	6.4 inches	13 MP + 5 MP + 2 MP Triple Rear & 8 MP Fro...	Memory C Supported, u 512
2	Samsung Galaxy A13	4.30	75	Dual Sim, 3G, 4G, VoLTE,	4 GB RAM	5000 mAh Battery	6.6 inches	50 MP Quad Rear & 8 MP Front Camera	Memory C Supported, up
3	Samsung Galaxy F23	4.10	73	Dual Sim, 3G, 4G, VoLTE,	4 GB RAM	6000 mAh Battery	6.4 inches	48 MP Quad Rear & 13 MP Front Camera	Memory C Supported, up
4	Samsung Galaxy A03s (4GB RAM + 64GB)	4.10	69	Dual Sim, 3G, 4G, VoLTE,	4 GB RAM	5000 mAh Battery	6.5 inches	13 MP + 2 MP + 2 MP Triple Rear & 5 MP Fro...	Memory C Supported, up
...	...	...	...	...	...	...	...	...	
1365	TCL 40R	4.05	75	Dual Sim, 3G, 4G, 5G, VoLTE,	4 GB RAM	5000 mAh Battery	6.6 inches	50 MP + 2 MP + 2 MP Triple Rear & 8 MP Fro...	Memory C (Hyb
1366	TCL 50 XL NxtPaper 5G	4.10	80	Dual Sim, 3G, 4G, VoLTE,	8 GB RAM	5000 mAh Battery	6.8 inches	50 MP + 2 MP Dual Rear & 16 MP Front Camera	Memory C (Hyb

	Name	Rating	Spec_score	No_of_sim	Ram	Battery	Display	Camera	External_Mem
1367	TCL 50 XE NxtPaper 5G	4.00	80	Dual Sim, 3G, 4G, 5G, VoLTE,	6 GB RAM	5000 mAh Battery	6.6 inches	50 MP + 2 MP Dual Rear & 16 MP Front Camera	Memory C Supported, up
1368	TCL 40 NxtPaper 5G	4.50	79	Dual Sim, 3G, 4G, 5G, VoLTE,	6 GB RAM	5000 mAh Battery	6.6 inches	50 MP + 2 MP + 2 MP Triple Rear & 8 MP Fro...	Memory C Supported, up
1369	TCL Trifold	4.65	93	Dual Sim, 3G, 4G, 5G, VoLTE, Vo5G,	12 GB RAM	4600 mAh Battery	10 inches	Foldable Display, Dual Display	50 MP + 48 M 8 MP Triple R & 32 MP

1370 rows × 17 columns

In [9]:
```
data['Price'] = data['Price'].str.replace(',','')
```

In [10]:
```
data["Price"]
```

Out[10]:
```
0         9999
1         9990
2        11999
3        11999
4        11999
         ...
1365     18999
1366     24990
1367     23990
1368     22499
1369    119990
Name: Price, Length: 1370, dtype: object
```

In [11]:
```
data.columns
```

Out[11]:
```
Index(['Name', 'Rating', 'Spec_score', 'No_of_sim', 'Ram', 'Battery',
       'Display', 'Camera', 'External_Memory', 'Android_version', 'Price',
       'company', 'Inbuilt_memory', 'fast_charging', 'Screen_resolution',
       'Processor', 'Processor_name'],
      dtype='object')
```

In [12]:
```
int_columns =['Ram','Battery','Display','External_Memory','Android_version',
              'Inbuilt_memory', 'fast_charging', 'Screen_resolution']
```

In [15]:
```
for column in int_columns:
    data[column] = data[column].astype(str).str.extract('(\d+)').astype(float
```

In [16]:
```
data[int_columns]
```

Out[16]:

	Ram	Battery	Display	External_Memory	Android_version	Inbuilt_memory	fast_charging
0	4.0	6000.0	6.0	1.0	13.0	128.0	25.0
1	2.0	4000.0	6.0	512.0	10.0	32.0	15.0
2	4.0	5000.0	6.0	1.0	12.0	64.0	25.0
3	4.0	6000.0	6.0	1.0	12.0	64.0	NaN
4	4.0	5000.0	6.0	1.0	11.0	64.0	15.0
...	...	...	...	...	...	...	...
1365	4.0	5000.0	6.0	NaN	12.0	64.0	15.0
1366	8.0	5000.0	6.0	NaN	14.0	128.0	33.0
1367	6.0	5000.0	6.0	1.0	13.0	256.0	18.0
1368	6.0	5000.0	6.0	1.0	13.0	256.0	15.0
1369	12.0	4600.0	10.0	50.0	13.0	256.0	67.0

1370 rows × 8 columns

In [17]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1370 entries, 0 to 1369
Data columns (total 17 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Name               1370 non-null   object 
 1   Rating             1370 non-null   float64
 2   Spec_score         1370 non-null   int64  
 3   No_of_sim          1370 non-null   object 
 4   Ram                1370 non-null   float64
 5   Battery            1370 non-null   float64
 6   Display            1370 non-null   float64
 7   Camera             1370 non-null   object 
 8   External_Memory    1078 non-null   float64
 9   Android_version    927 non-null    float64
 10  Price              1370 non-null   object 
 11  company            1370 non-null   object 
 12  Inbuilt_memory     1350 non-null   float64
 13  fast_charging      1241 non-null   float64
 14  Screen_resolution  1367 non-null   float64
 15  Processor          1342 non-null   object 
 16  Processor_name     1370 non-null   object 
dtypes: float64(9), int64(1), object(7)
memory usage: 182.1+ KB
```

In [18]:
```
for i in int_columns:
    data[i] = data[i].fillna(data[i].mode()[0])
```

In [20]:
```
cat_features = data.select_dtypes(include='object').columns
```

In [21]:
```
from sklearn.preprocessing import LabelEncoder
```

In [22]:
```
le = LabelEncoder()
```

In [23]:
```
for i in cat_features:
    data[i] = le.fit_transform(data[i])
```

In [24]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1370 entries, 0 to 1369
Data columns (total 17 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Name               1370 non-null   int32  
 1   Rating             1370 non-null   float64
 2   Spec_score         1370 non-null   int64  
 3   No_of_sim          1370 non-null   int32  
 4   Ram                1370 non-null   float64
 5   Battery            1370 non-null   float64
 6   Display            1370 non-null   float64
 7   Camera             1370 non-null   int32  
 8   External_Memory    1370 non-null   float64
 9   Android_version    1370 non-null   float64
 10  Price              1370 non-null   int32  
 11  company            1370 non-null   int32  
 12  Inbuilt_memory     1370 non-null   float64
 13  fast_charging      1370 non-null   float64
 14  Screen_resolution  1370 non-null   float64
 15  Processor          1370 non-null   int32  
 16  Processor_name     1370 non-null   int32  
dtypes: float64(9), int32(7), int64(1)
memory usage: 144.6 KB
```

In [25]:
```
from sklearn.model_selection import train_test_split
```

In [28]:
```
x = data.drop('Price',axis =1)
```

In [29]:
```
y = data['Price']
```

In [30]:
```
X_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.2)
```

In [31]:
```
from sklearn.linear_model import LinearRegression
```

In [32]:
```
model = LinearRegression()
```

In [33]:
```
model.fit(X_train,y_train)
```

Out[33]:
```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [34]:
```
y_pred = model.predict(X_test)
```

In [35]:
```
y_pred
```

Out[35]:
```
array([191.23323886, 235.56329098, 283.48213735, 240.33619565,
       196.60688623, 288.32083711, 279.60874337, 237.54759636,
       256.33503341, 216.46427984, 221.64358181, 208.89084974,
       206.42099103, 201.53649832, 240.47418375, 243.48698619,
       289.63424849, 216.18227426, 235.1923605 , 214.77934187,
       215.09655348, 213.81569053, 217.32733137, 175.07180298,
       216.95334964, 210.54957474, 297.15608298, 262.88634982,
       178.96435508, 199.83947098, 138.6638806 , 263.82309576,
       187.2106469 , 230.41991579, 263.65092352, 151.79189868,
       167.19743935, 271.92495664, 203.08712607, 191.87549254,
       212.91177426, 249.50436527, 194.24892297, 192.90582606,
       225.06724744, 214.75471259, 224.90986629, 201.45426361,
       234.28689067, 245.51349811, 221.54575418, 287.5473451 ,
       248.79451128, 266.01915471, 218.03092118, 272.99117415,
       264.17373355, 344.45188079, 208.32589562, 246.58344572,
       210.55548679, 274.49150568, 220.29272498, 207.16436416,
       292.79712381, 241.29412467, 250.34129789, 181.47613024,
       206.91958523, 218.17203422, 264.94148607, 195.91049024,
       254.63446391, 251.56324416, 253.93270467, 256.27637225,
       260.66100767, 173.05214752, 173.8213681 , 200.43896516,
       192.04060018, 226.14432781, 144.77495899, 242.23815456,
       185.51124111, 229.27429434, 215.53459396, 204.93627012,
       233.9529173 , 219.74315879, 181.55650953, 187.8643348 ,
       275.00920884, 305.80319448, 258.58122498, 278.4797636 ,
       191.4540919 , 300.71580356, 167.58314837, 218.82181804,
       251.63236958, 295.57751269, 164.79913014, 310.05914749,
       259.56433853, 174.41092498, 226.16291323, 211.64893218,
       200.88647128, 268.46262966, 159.95843666, 204.16281457,
       259.79094028, 225.63795121, 176.1499813 , 296.43532813,
       213.70726216, 322.59188172, 139.72472835, 165.29045405,
       271.64616997, 256.26896395, 262.06341046, 312.87472621,
       310.47090741, 247.35917407, 192.77896722, 251.77112919,
       282.53400645, 204.97648749, 161.17745234, 204.60433203,
       304.05834651, 214.23888703, 217.33182955, 223.11316826,
       260.10824102, 181.36258368, 245.4714249 , 241.32295082,
       229.34250264, 199.42013763, 260.22608438, 211.52662304,
       308.93744814, 215.5645255 , 188.74900077, 178.01025806,
       158.10591479, 203.53405479, 295.71975071, 256.07590769,
       249.18348126, 233.8059848 , 264.68053295, 235.61496953,
       261.16992795, 174.06958461, 289.71466846, 212.66935277,
       246.11171693, 292.68842871, 222.76055545, 251.71501512,
       264.26134931, 285.48183727, 216.36211232, 211.79150204,
       252.41187153, 181.14111314, 277.55232433, 234.06756176,
       214.37889827, 215.90902851, 195.71391773, 221.07615321,
       218.52777738, 191.94637993, 229.40533636, 175.79285109,
       209.6649562 , 224.18137173, 219.89486846, 193.50921932,
       233.23516067, 270.42505534, 209.89921069, 209.3372742 ,
       202.76046899, 241.26844218, 312.78147563, 216.3123451 ,
       243.60332426, 222.70646281, 165.81066313, 241.86438369,
       243.15003153, 206.78761661, 238.78625499, 250.71922923,
       245.28989343, 216.68331982, 197.29733614, 185.15721712,
       157.5127734 , 296.79769323, 152.84365022, 218.03588333,
       169.33073048, 286.42938781, 208.42772072, 299.51741428,
       229.46899839, 261.2848569 , 204.97858269, 281.396314  ,
       230.04969859, 168.15484827, 228.06089582, 198.72087206,
       239.9566485 , 172.45891114, 247.47848516, 233.20362494,
       199.48068482, 183.75858762, 214.92872446, 273.12441587,
```

```
211.49087803, 244.47806307, 245.55202713, 310.35680617,
226.2058541 , 221.89121296, 285.59356064, 268.8356516 ,
221.28754219, 127.24177479, 252.359254  , 236.63719735,
258.81137029, 221.34821455, 138.65453986, 282.59270363,
299.02867849, 298.56304583, 330.31015939, 217.66223137,
225.00832284, 218.10623538, 174.43377477, 229.36667109,
247.80358621, 322.02438319, 191.53321791, 259.96965867,
189.31770649, 282.77511133, 195.40919137, 254.68196076,
222.32465254, 230.67762259, 174.31238346, 222.69978623,
205.91758693, 177.78368981, 191.69830204, 235.2317627 ,
340.01238575, 165.93855525, 214.65934647, 229.78962615,
191.37961304, 214.98241538])
```

In [36]:
```
from sklearn.metrics import r2_score
```

In [37]:
```
r2_score(y_test,y_pred)
```

Out[37]:
```
0.044075879755962166
```

In [ ]:
```
r2_score()
```