In [283…
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [285…
```
df = pd.read_csv('laptop_data.csv')
```

In [286…
```
df.head()
```

Out[286]:

	Unnamed: 0	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	Op
0	0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	mac
1	1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	mac
2	2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	No
3	3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	AMD Radeon Pro 455	mac
4	4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	Intel Iris Plus Graphics 650	mac

In [287…
```
df.shape
```

Out[287]:
```
(1303, 12)
```

In [288…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Unnamed: 0        1303 non-null   int64  
 1   Company           1303 non-null   object 
 2   TypeName          1303 non-null   object 
 3   Inches            1303 non-null   float64
 4   ScreenResolution  1303 non-null   object 
 5   Cpu               1303 non-null   object 
 6   Ram               1303 non-null   object 
 7   Memory            1303 non-null   object 
 8   Gpu               1303 non-null   object 
 9   OpSys             1303 non-null   object 
 10  Weight            1303 non-null   object 
 11  Price             1303 non-null   float64
dtypes: float64(2), int64(1), object(9)
memory usage: 122.3+ KB
```

In [289…]
```
df.duplicated().sum()
```

Out[289]:
```
0
```

In [290…]
```
df.isnull().sum()
```

Out[290]:
```
Unnamed: 0          0
Company             0
TypeName            0
Inches              0
ScreenResolution    0
Cpu                 0
Ram                 0
Memory              0
Gpu                 0
OpSys               0
Weight              0
Price               0
dtype: int64
```

In [291…]
```
df.drop(columns=['Unnamed: 0'],inplace=True)
```

In [292…]
```
df.head()
```

Out[292]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	No OS	1.86kg
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	AMD Radeon Pro 455	macOS	1.83kg
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37kg

In [293…

```
df['Ram'] = df['Ram'].str.replace('GB','')
df['Weight'] = df['Weight'].str.replace('kg','')
```

In [294…

```
df.head()
```

Out[294]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37

In [295…

```
df['Ram'] = df['Ram'].astype('int32')
df['Weight'] = df['Weight'].astype('float32')
```

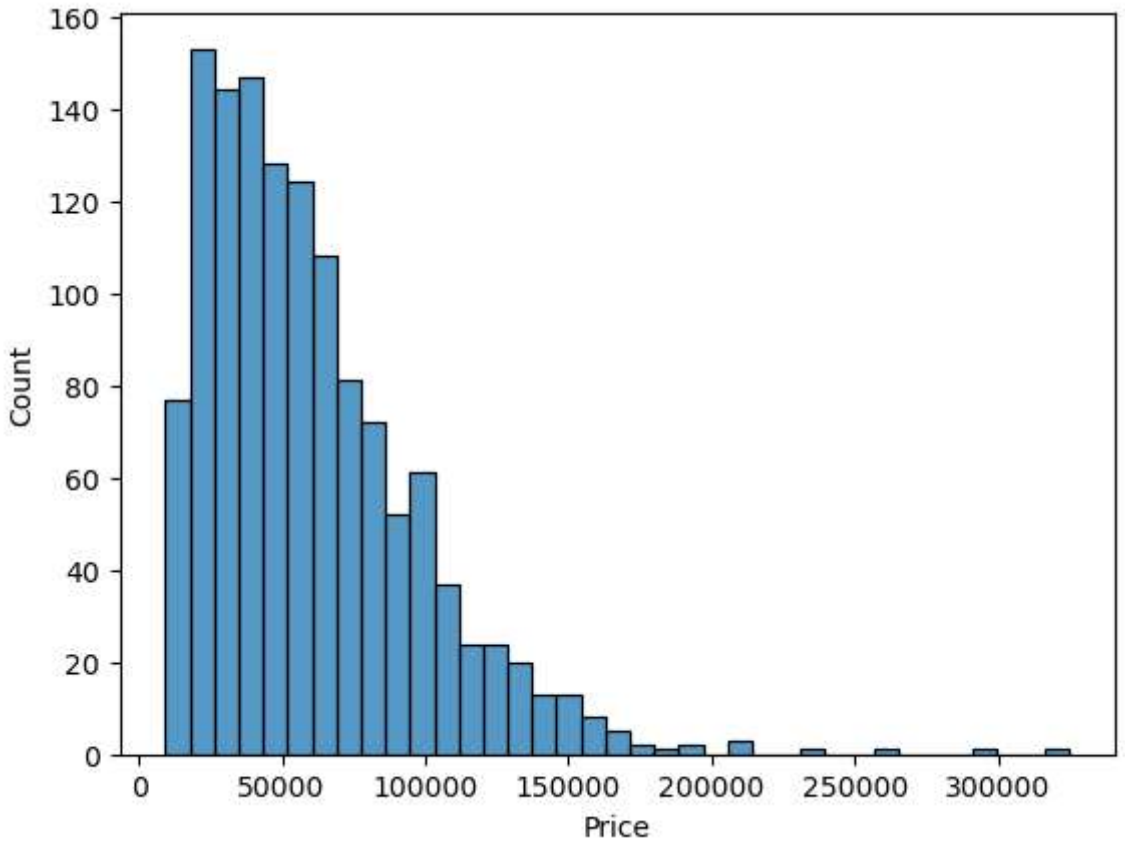
In [296…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Company           1303 non-null   object
 1   TypeName          1303 non-null   object
 2   Inches            1303 non-null   float64
 3   ScreenResolution  1303 non-null   object
 4   Cpu               1303 non-null   object
 5   Ram               1303 non-null   int32
 6   Memory            1303 non-null   object
 7   Gpu               1303 non-null   object
 8   OpSys             1303 non-null   object
 9   Weight            1303 non-null   float32
 10  Price             1303 non-null   float64
dtypes: float32(1), float64(2), int32(1), object(7)
memory usage: 101.9+ KB
```

In [297…
```
import seaborn as sns
```

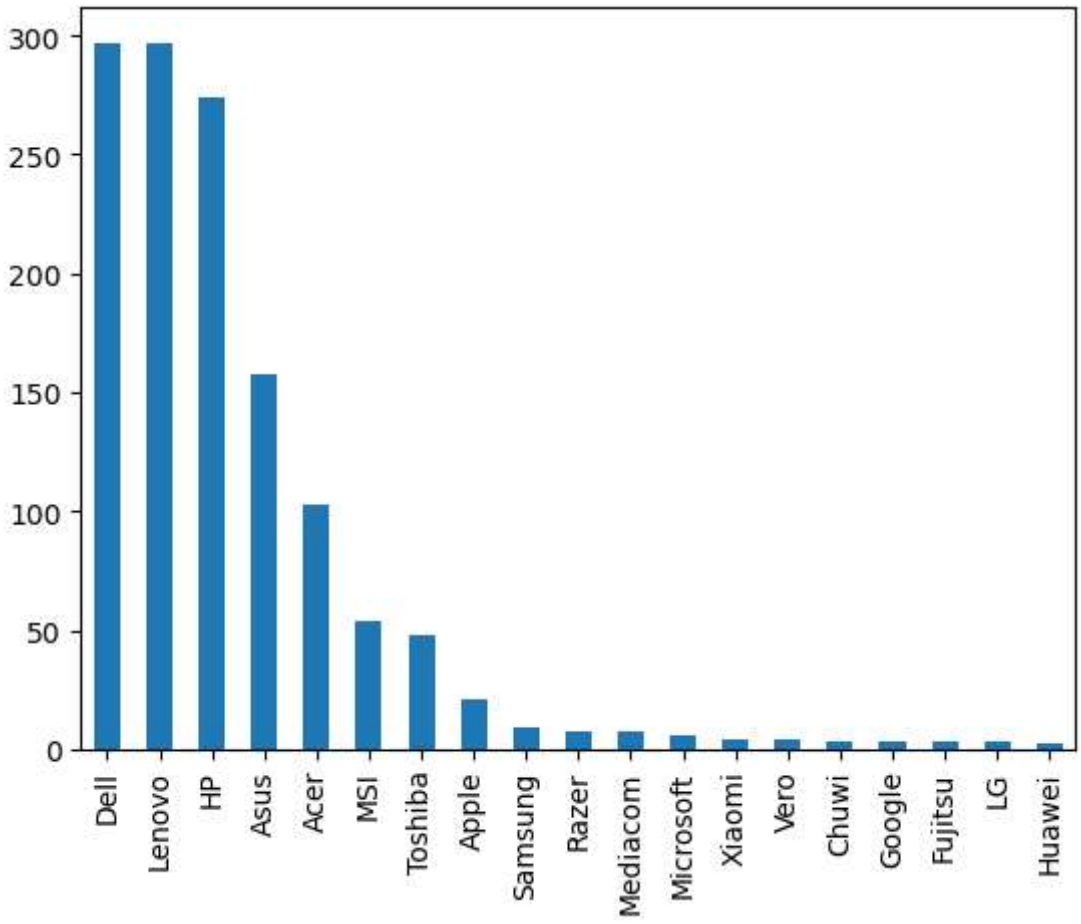
In [298…
```
sns.histplot(df['Price'])
```

Out[298]:
```
<Axes: xlabel='Price', ylabel='Count'>
```



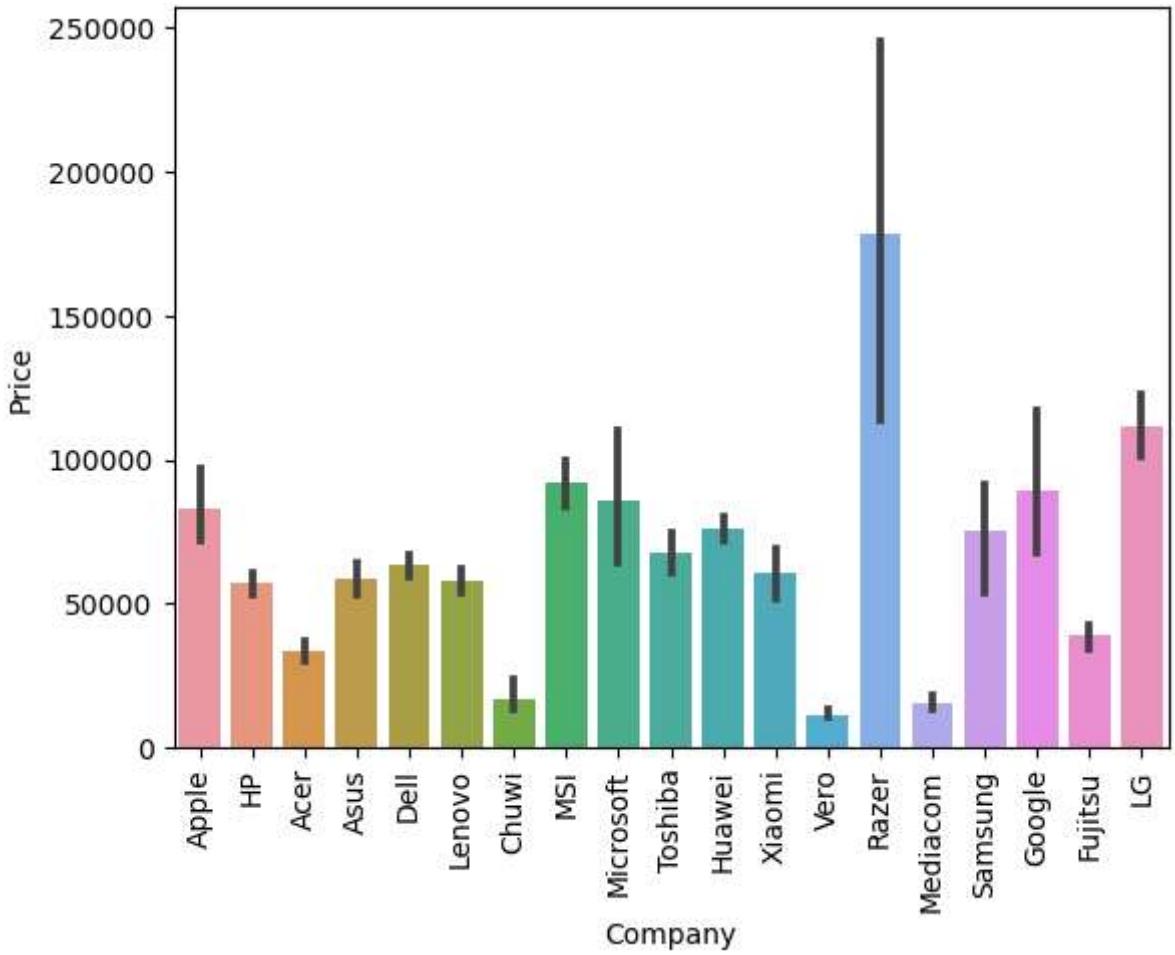
In [299…
```
df['Company'].value_counts().plot(kind='bar')
```

Out[299]:
```
<Axes: >
```



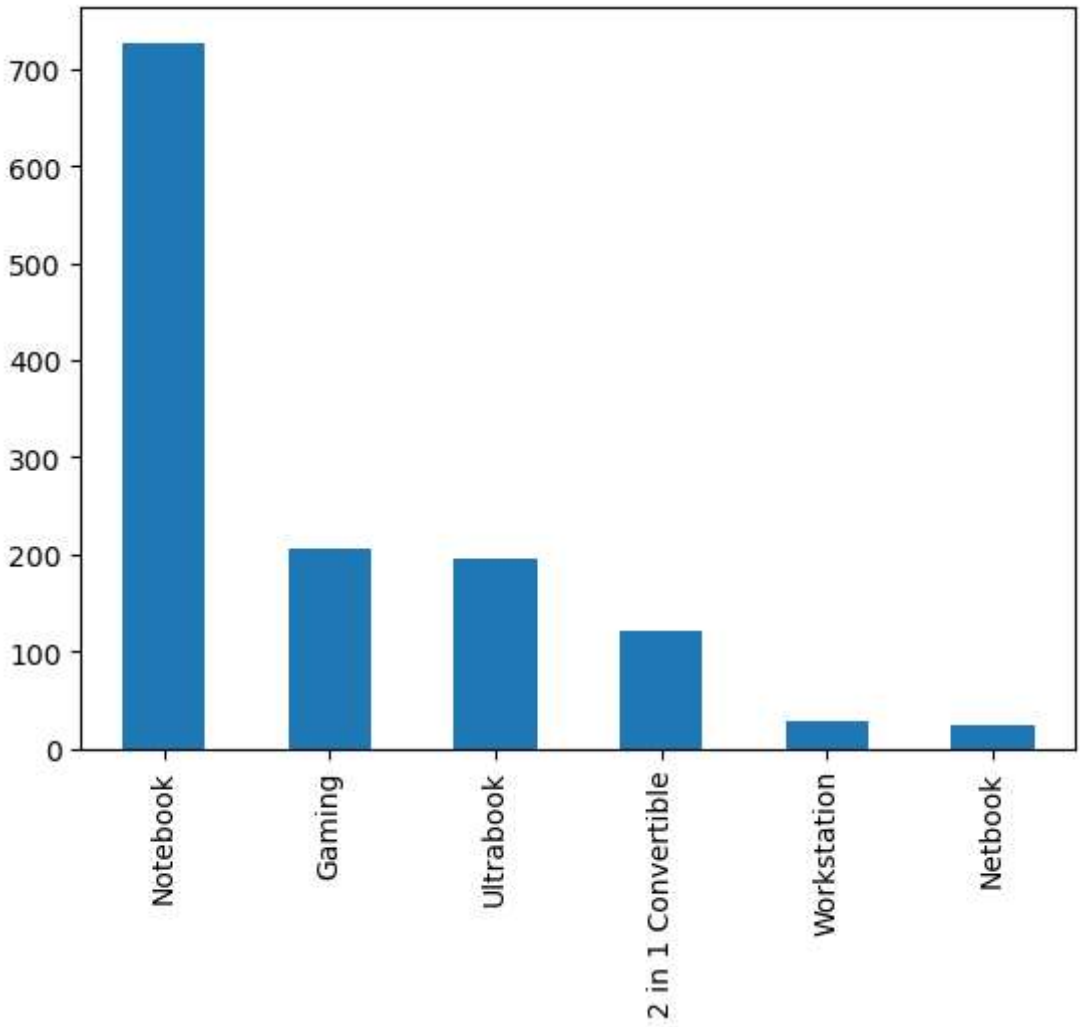
In [300…

```
sns.barplot(x=df['Company'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



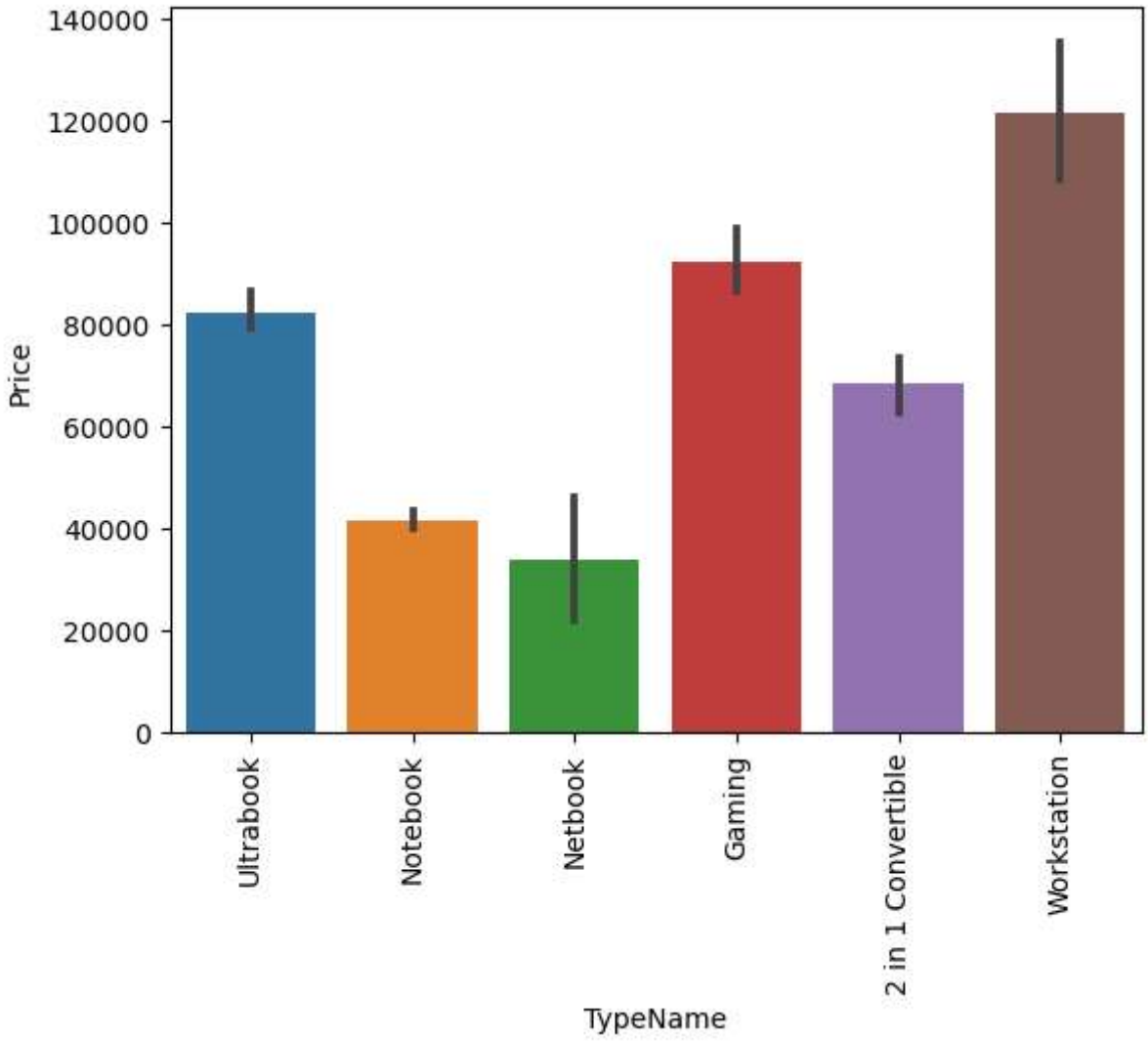
In [301…
```
df['TypeName'].value_counts().plot(kind='bar')
```

Out[301]: <Axes: >



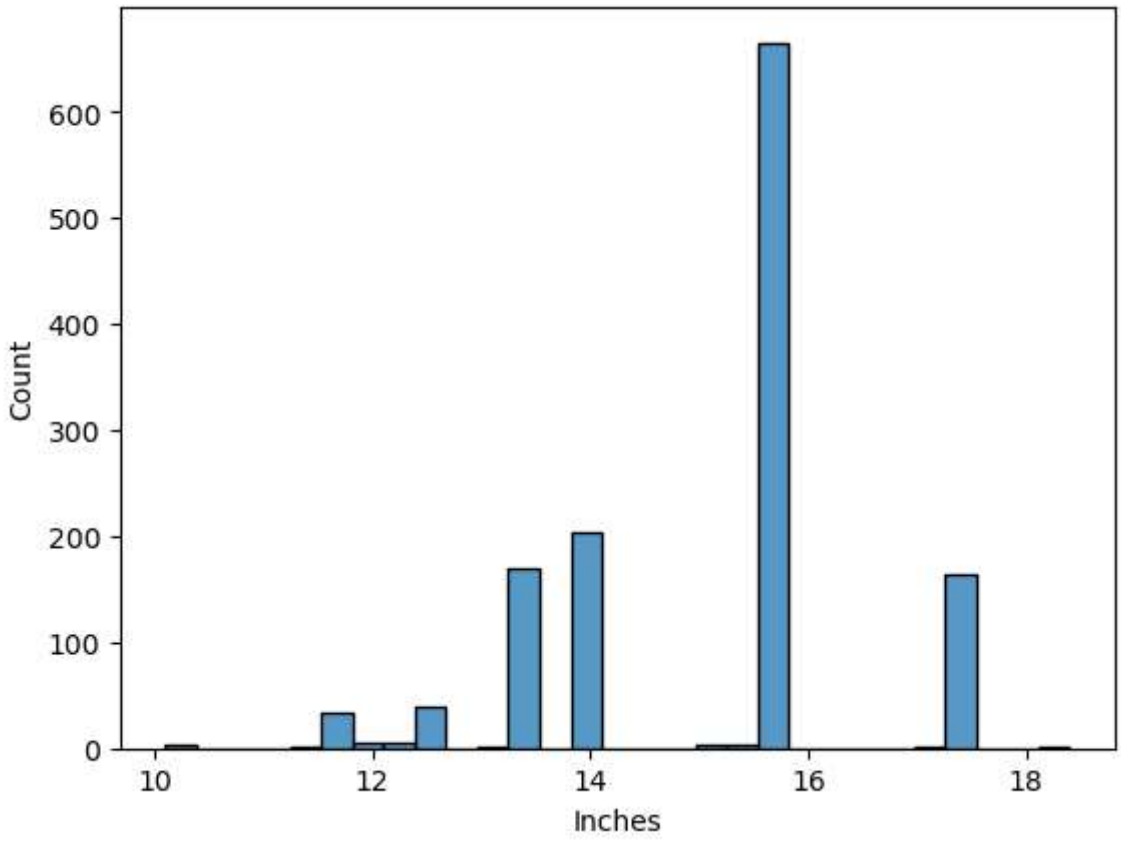
In [302…

```
sns.barplot(x=df['TypeName'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



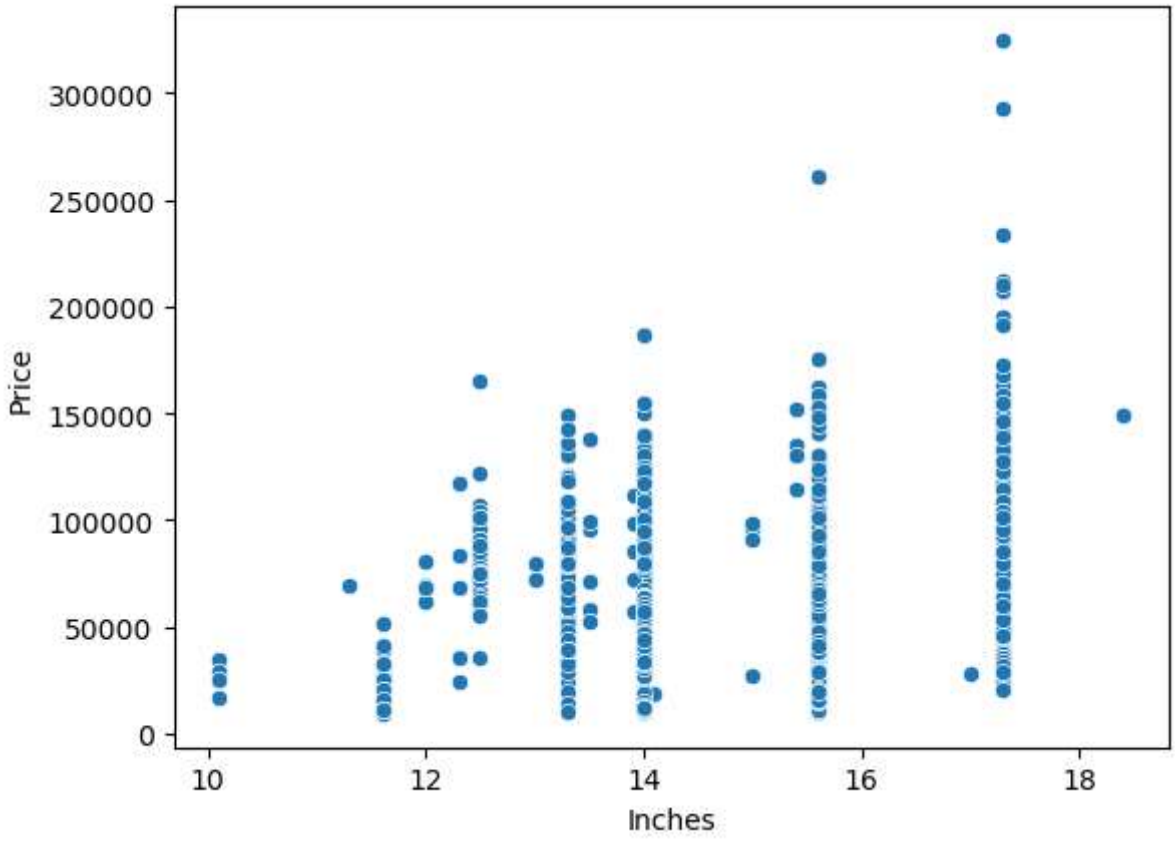
In [303…
```
sns.histplot(df['Inches'])
```

Out[303]:
```
<Axes: xlabel='Inches', ylabel='Count'>
```



In [304…
```
sns.scatterplot(x=df['Inches'],y=df['Price'])
```

Out[304]:
```
<Axes: xlabel='Inches', ylabel='Price'>
```



In [305…
```
df['ScreenResolution'].value_counts()
```

Out[305]:
```
Full HD 1920x1080                                   507
1366x768                                            281
IPS Panel Full HD 1920x1080                         230
IPS Panel Full HD / Touchscreen 1920x1080            53
Full HD / Touchscreen 1920x1080                      47
1600x900                                             23
Touchscreen 1366x768                                 16
Quad HD+ / Touchscreen 3200x1800                     15
IPS Panel 4K Ultra HD 3840x2160                      12
IPS Panel 4K Ultra HD / Touchscreen 3840x2160        11
4K Ultra HD / Touchscreen 3840x2160                  10
4K Ultra HD 3840x2160                                 7
Touchscreen 2560x1440                                 7
IPS Panel 1366x768                                    7
IPS Panel Quad HD+ / Touchscreen 3200x1800            6
IPS Panel Retina Display 2560x1600                    6
IPS Panel Retina Display 2304x1440                    6
Touchscreen 2256x1504                                 6
IPS Panel Touchscreen 2560x1440                       5
IPS Panel Retina Display 2880x1800                    4
IPS Panel Touchscreen 1920x1200                       4
1440x900                                              4
IPS Panel 2560x1440                                   4
IPS Panel Quad HD+ 2560x1440                          3
Quad HD+ 3200x1800                                    3
1920x1080                                             3
Touchscreen 2400x1600                                 3
2560x1440                                             3
IPS Panel Touchscreen 1366x768                        3
IPS Panel Touchscreen / 4K Ultra HD 3840x2160         2
IPS Panel Full HD 2160x1440                           2
IPS Panel Quad HD+ 3200x1800                          2
IPS Panel Retina Display 2736x1824                    1
IPS Panel Full HD 1920x1200                           1
IPS Panel Full HD 2560x1440                           1
IPS Panel Full HD 1366x768                            1
Touchscreen / Full HD 1920x1080                       1
Touchscreen / Quad HD+ 3200x1800                      1
Touchscreen / 4K Ultra HD 3840x2160                   1
IPS Panel Touchscreen 2400x1600                       1
Name: ScreenResolution, dtype: int64
```

In [306…
```
df['Touchscreen'] = df['ScreenResolution'].apply(lambda x:1 if 'Touchscreen' in x else
```

In [307…
```
df.sample(5)
```

Out[307]:

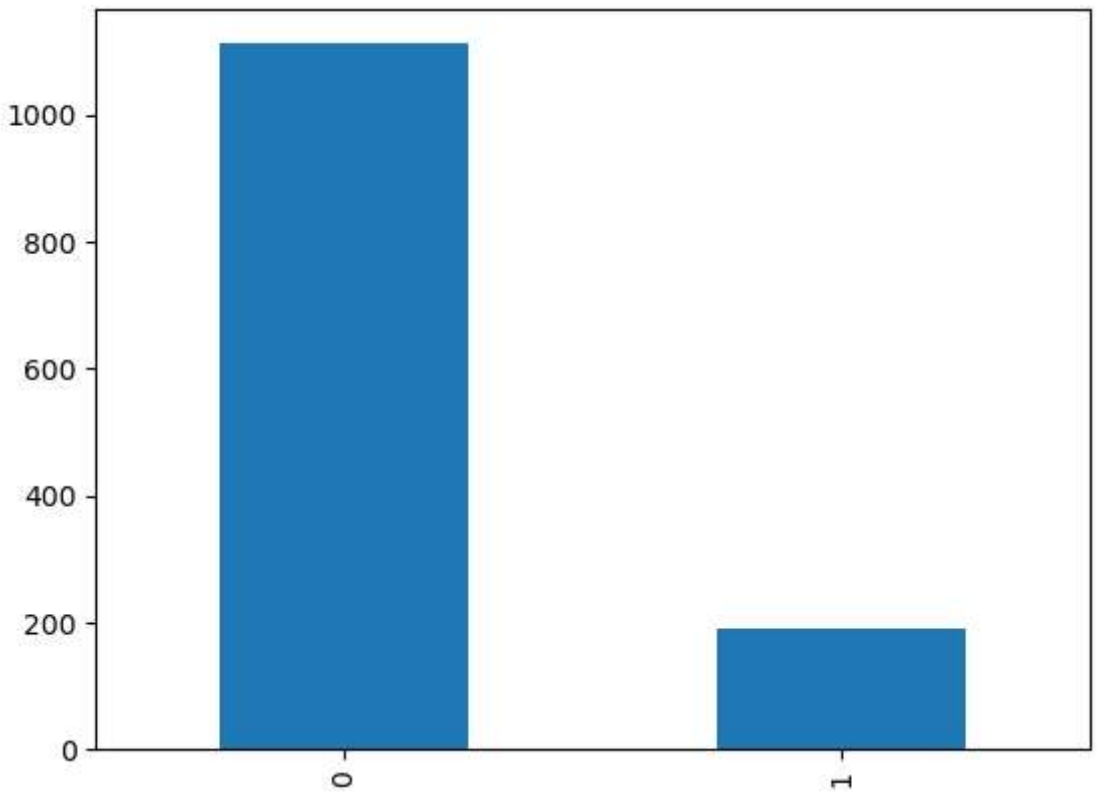
	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	W
291	Asus	Gaming	17.3	Full HD 1920x1080	Intel Core i7 7700HQ 2.8GHz	8	1TB HDD	Nvidia GeForce GTX 1050	Windows 10	
808	Dell	Gaming	15.6	4K Ultra HD 3840x2160	Intel Core i7 7700HQ 2.8GHz	16	512GB SSD	Nvidia GeForce GTX 1050 Ti	Windows 10	
489	Asus	Ultrabook	12.5	Full HD 1920x1080	Intel Core i7 7500U 2.7GHz	16	512GB SSD	Intel HD Graphics 620	Windows 10	
89	Dell	Ultrabook	13.3	IPS Panel Full HD 1920x1080	Intel Core i7 8550U 1.8GHz	8	256GB SSD	Intel UHD Graphics 620	Windows 10	
151	Dell	Gaming	15.6	Full HD 1920x1080	Intel Core i7 7700HQ 2.8GHz	8	1.0TB Hybrid	Nvidia GeForce GTX 1050	Windows 10	

In [308…

```
df['Touchscreen'].value_counts().plot(kind='bar')
```

Out[308]:

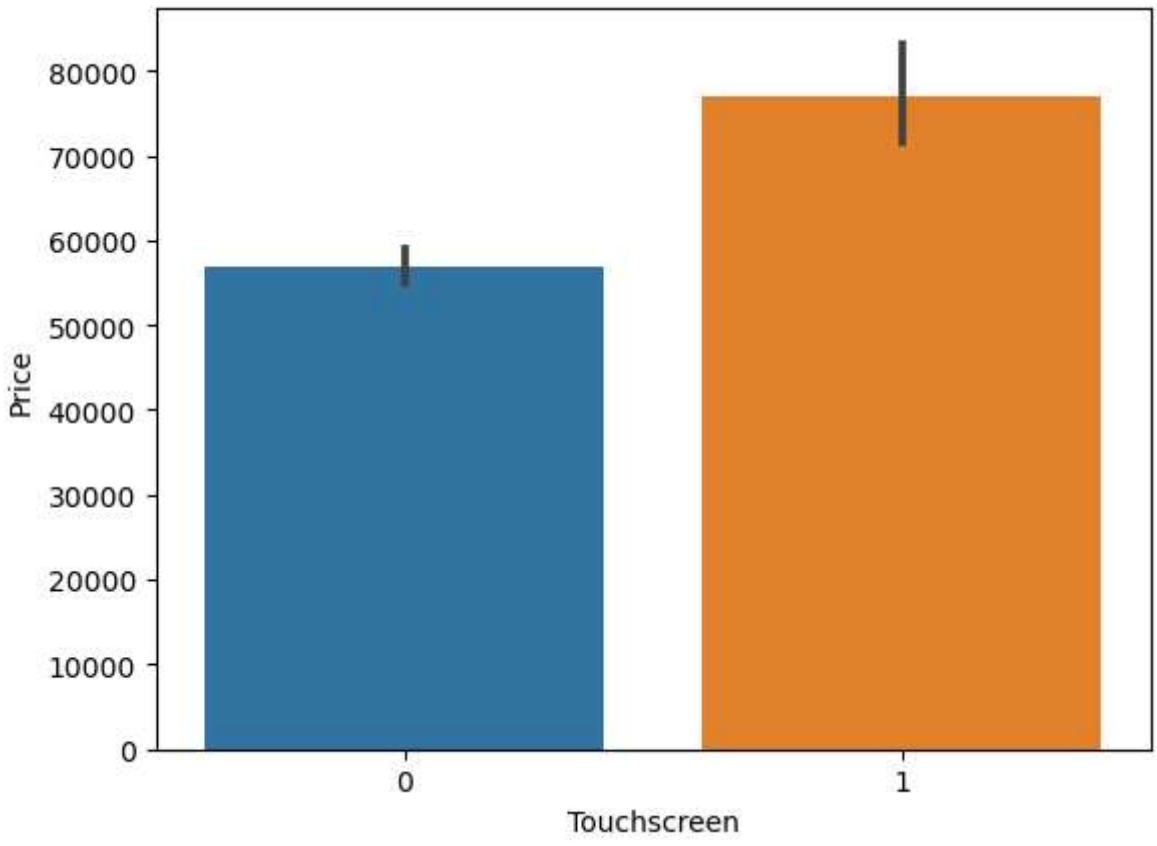
```
<Axes: >
```



In [309…

```
sns.barplot(x=df['Touchscreen'],y=df['Price'])
```

Out[309]: <Axes: xlabel='Touchscreen', ylabel='Price'>



In [310…
```
df['Ips'] = df['ScreenResolution'].apply(lambda x:1 if 'IPS' in x else 0)
```

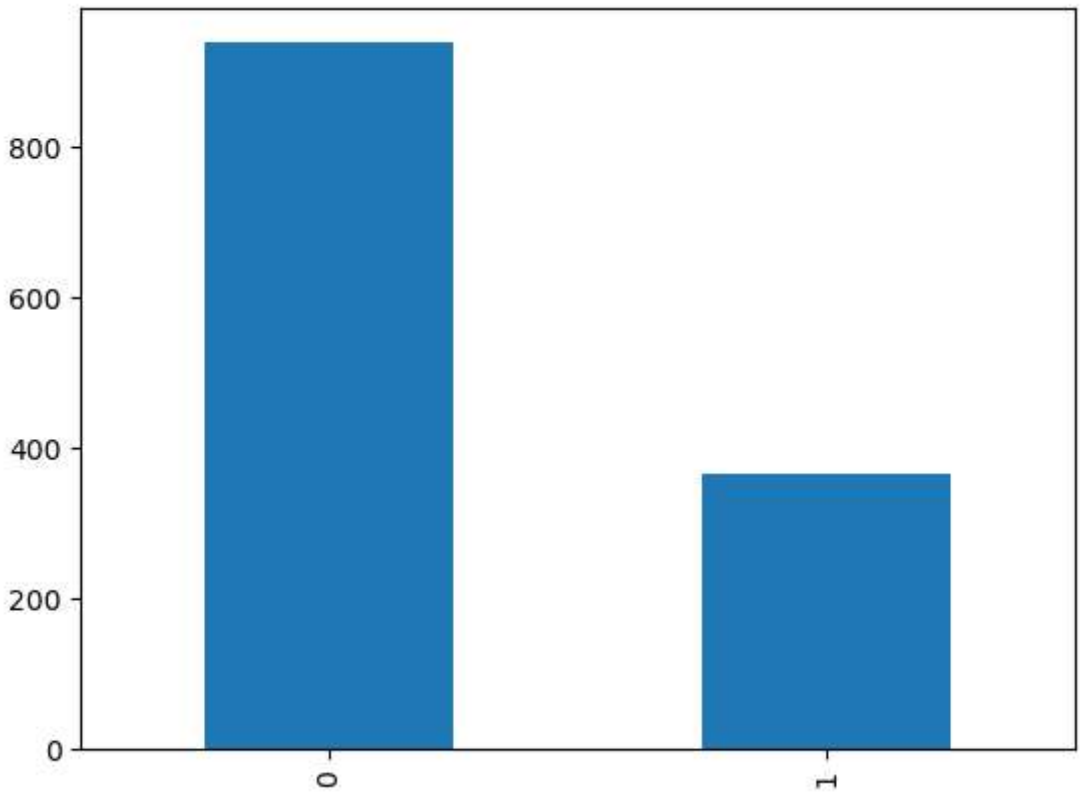
In [311…
```
df.head()
```

Out[311]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37

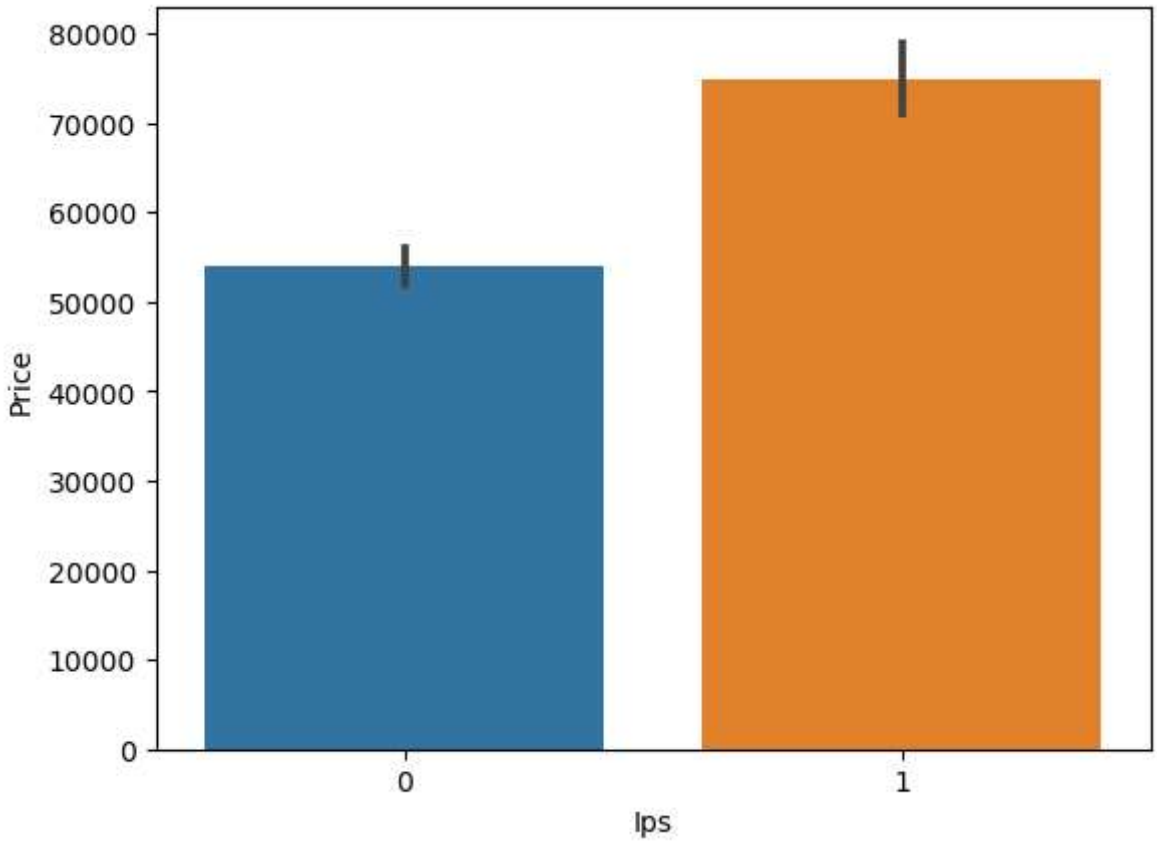
In [312…
```
df['Ips'].value_counts().plot(kind='bar')
```

Out[312]:
```
<Axes: >
```



In [313…
```
sns.barplot(x=df['Ips'],y=df['Price'])
```

Out[313]:
```
<Axes: xlabel='Ips', ylabel='Price'>
```



In [314…
```
new = df['ScreenResolution'].str.split('x',n=1,expand=True)
```

In [315…
```
df['X_res'] = new[0]
df['Y_res'] = new[1]
```

In [316…
```
df.sample(5)
```

Out[316]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	W
587	Lenovo	Ultrabook	14.0	IPS Panel Full HD 1920x1080	Intel Core i7 6600U 2.6GHz	12	256GB SSD	Intel HD Graphics 520	Windows 10	
184	Xiaomi	Notebook	15.6	IPS Panel Full HD 1920x1080	Intel Core i5 8250U 1.6GHz	8	256GB SSD	Nvidia GeForce MX150	No OS	
420	Lenovo	2 in 1 Convertible	15.6	IPS Panel 4K Ultra HD / Touchscreen 3840x2160	Intel Core i7 7700HQ 2.8GHz	16	512GB SSD	Nvidia GeForce GTX 1050	Windows 10	
603	MSI	Gaming	17.3	Full HD 1920x1080	Intel Core i7 6820HK 2.7GHz	16	128GB SSD + 1TB HDD	Nvidia GeForce GTX 970M	Windows 10	
1045	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 6300U 2.4GHz	8	256GB SSD + 500GB HDD	Intel HD Graphics 520	Windows 10	

In [317…

```
df['X_res'] = df['X_res'].str.replace(',','').str.findall(r'(\d+\.?\d+)').apply(lambda
```

In [318…

```
df.head()
```

Out[318]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37

In [319…
```
df['X_res'] = df['X_res'].astype('int')
df['Y_res'] = df['Y_res'].astype('int')
```

In [320…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 15 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Company           1303 non-null   object 
 1   TypeName          1303 non-null   object 
 2   Inches            1303 non-null   float64
 3   ScreenResolution  1303 non-null   object 
 4   Cpu               1303 non-null   object 
 5   Ram               1303 non-null   int32  
 6   Memory            1303 non-null   object 
 7   Gpu               1303 non-null   object 
 8   OpSys             1303 non-null   object 
 9   Weight            1303 non-null   float32
 10  Price             1303 non-null   float64
 11  Touchscreen       1303 non-null   int64  
 12  Ips               1303 non-null   int64  
 13  X_res             1303 non-null   int32  
 14  Y_res             1303 non-null   int32  
dtypes: float32(1), float64(2), int32(3), int64(2), object(7)
memory usage: 132.5+ KB
```

In [321…
```
df.corr(numeric_only = True)['Price']
```

Out[321]:
```
Inches         0.068197
Ram            0.743007
Weight         0.210370
Price          1.000000
Touchscreen    0.191226
Ips            0.252208
X_res          0.556529
Y_res          0.552809
Name: Price, dtype: float64
```

In [322…
```
df['ppi'] = (((df['X_res']**2) + (df['Y_res']**2))**0.5/df['Inches']).astype('float')
```

In [323…
```
df.corr(numeric_only = True)['Price']
```

Out[323]:
```
Inches         0.068197
Ram            0.743007
Weight         0.210370
Price          1.000000
Touchscreen    0.191226
Ips            0.252208
X_res          0.556529
Y_res          0.552809
ppi            0.473487
Name: Price, dtype: float64
```

In [324…
```
df.drop(columns=['ScreenResolution'],inplace=True)
```

In [325…
```
df.head()
```

Out[325]:

	Company	TypeName	Inches	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	Tou
0	Apple	Ultrabook	13.3	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	
1	Apple	Ultrabook	13.3	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	
2	HP	Notebook	15.6	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	
3	Apple	Ultrabook	15.4	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	
4	Apple	Ultrabook	13.3	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	

In [326…

```
df.drop(columns=['Inches','X_res','Y_res'],inplace=True)
```

In [327…

```
df.head()
```

Out[327]:

	Company	TypeName	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	Touchscreen
0	Apple	Ultrabook	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0
1	Apple	Ultrabook	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0
2	HP	Notebook	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0
3	Apple	Ultrabook	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0
4	Apple	Ultrabook	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0

In [328…

```
df['Cpu'].value_counts()
```

Out[328]:
```
Intel Core i5 7200U 2.5GHz       190
Intel Core i7 7700HQ 2.8GHz      146
Intel Core i7 7500U 2.7GHz       134
Intel Core i7 8550U 1.8GHz        73
Intel Core i5 8250U 1.6GHz        72
                                 ...
Intel Core M M3-6Y30 0.9GHz        1
AMD A9-Series 9420 2.9GHz          1
Intel Core i3 6006U 2.2GHz         1
AMD A6-Series 7310 2GHz            1
Intel Xeon E3-1535M v6 3.1GHz      1
Name: Cpu, Length: 118, dtype: int64
```

In [329…
```
df['Cpu Name'] = df['Cpu'].apply(lambda x:" ".join(x.split()[0:3]))
```

In [330…
```
df.head()
```

Out[330]:

	Company	TypeName	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	Touchscreen
0	Apple	Ultrabook	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0
1	Apple	Ultrabook	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0
2	HP	Notebook	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0
3	Apple	Ultrabook	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0
4	Apple	Ultrabook	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0

In [331…
```
def fetch_processor(text):
    if text == 'Intel Core i7' or text == 'Intel Core i5' or text == 'Intel Core i3':
        return text
    else:
        if text.split()[0] == 'Intel':
            return 'Other Intel Processor'
        else:
            return 'AMD Processor'
```

In [332…
```
df['Cpu brand'] = df['Cpu Name'].apply(fetch_processor)
```

In [333…
```
df.head()
```

Out[333]:

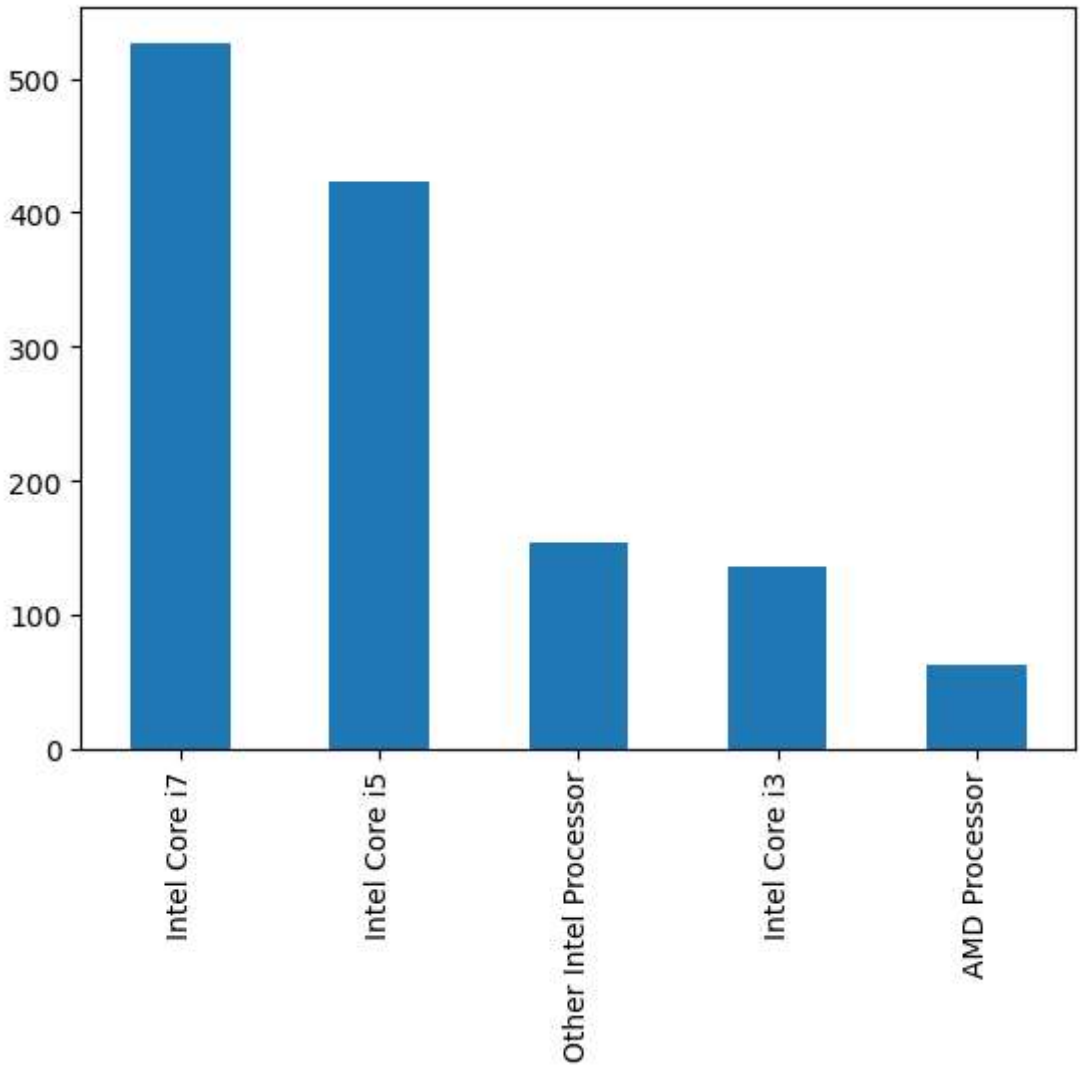
	Company	TypeName	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	Touchscreen
0	Apple	Ultrabook	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0
1	Apple	Ultrabook	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0
2	HP	Notebook	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0
3	Apple	Ultrabook	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0
4	Apple	Ultrabook	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0

In [334…

```
df['Cpu brand'].value_counts().plot(kind='bar')
```

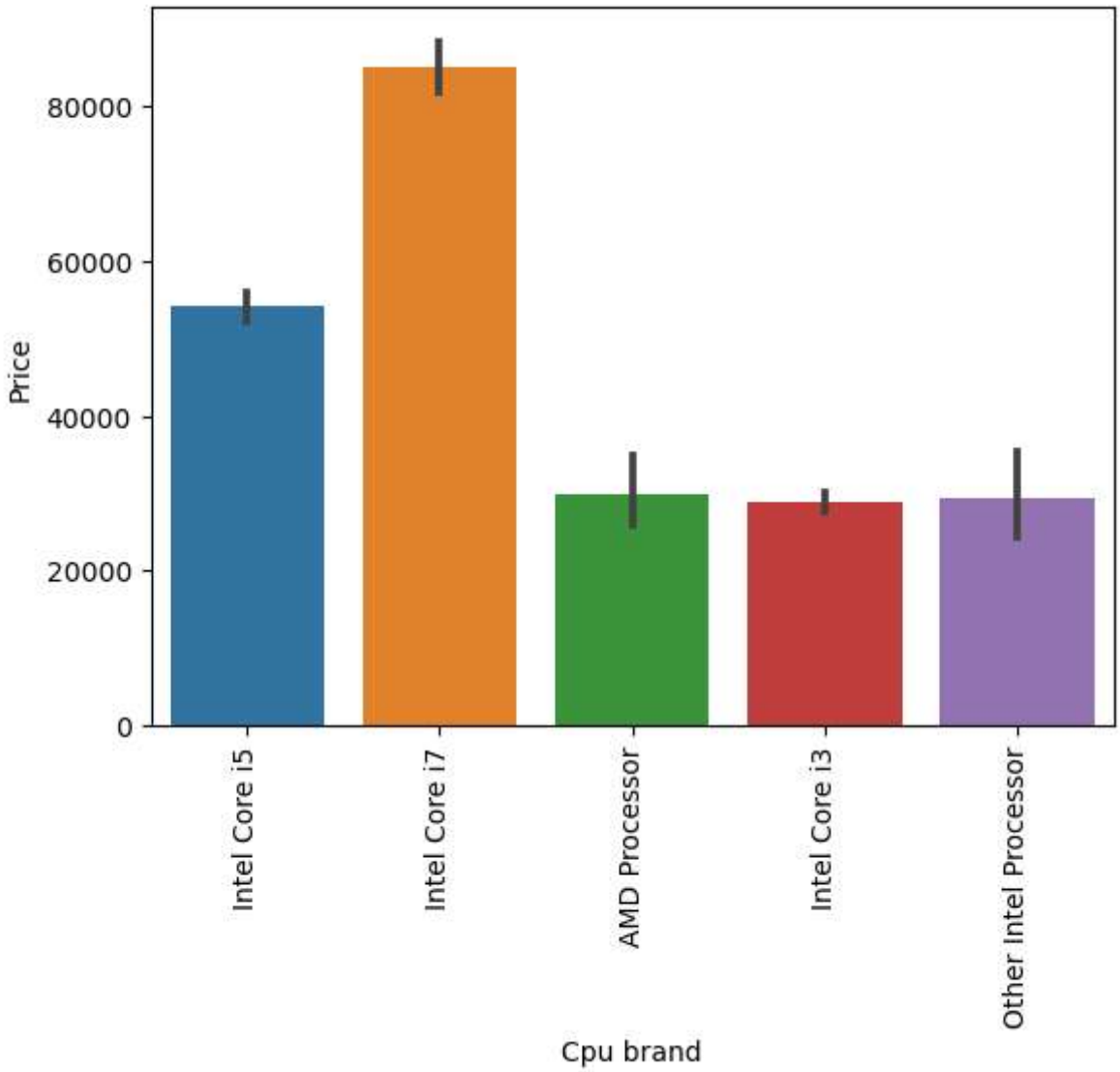
Out[334]:

```
<Axes: >
```



In [335…

```
sns.barplot(x=df['Cpu brand'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



In [336…
```
df.drop(columns=['Cpu','Cpu Name'],inplace=True)
```

In [337…
```
df.head()
```

Out[337]:

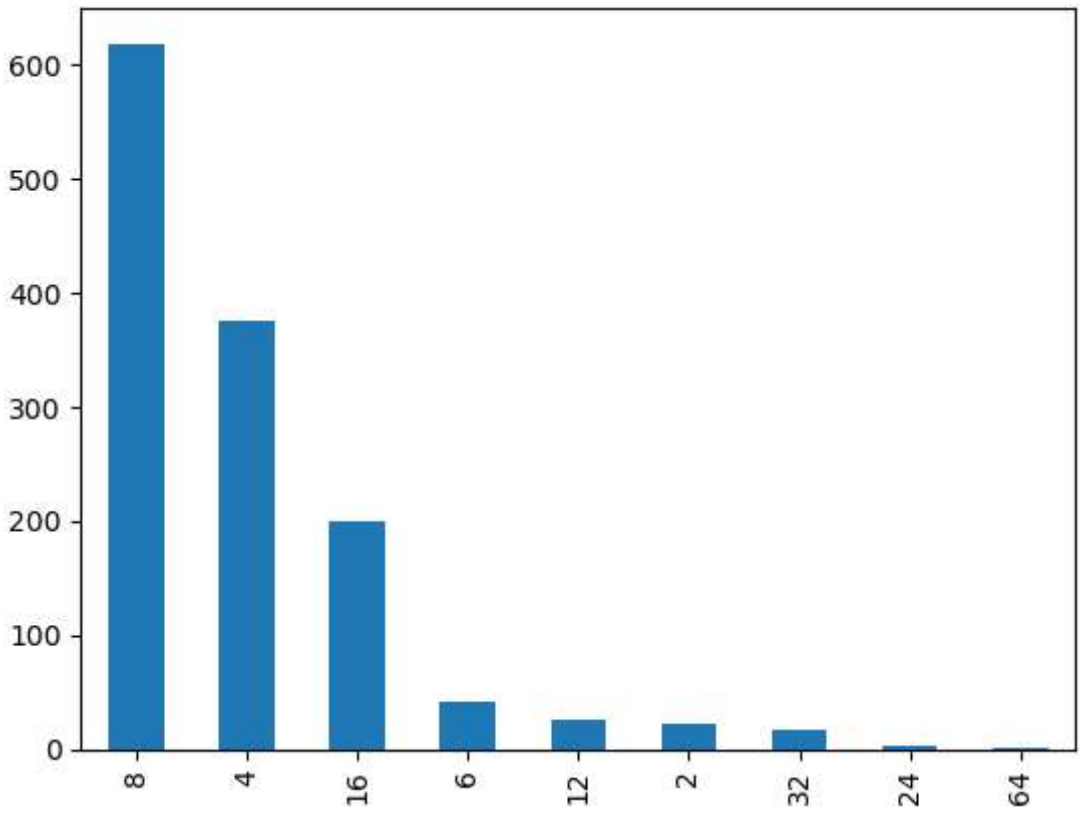
	Company	TypeName	Ram	Memory	Gpu	OpSys	Weight	Price	Touchscreen	Ips	
0	Apple	Ultrabook	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	2
1	Apple	Ultrabook	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	1
2	HP	Notebook	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	1
3	Apple	Ultrabook	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	2
4	Apple	Ultrabook	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1	2

In [338…

```
df['Ram'].value_counts().plot(kind='bar')
```

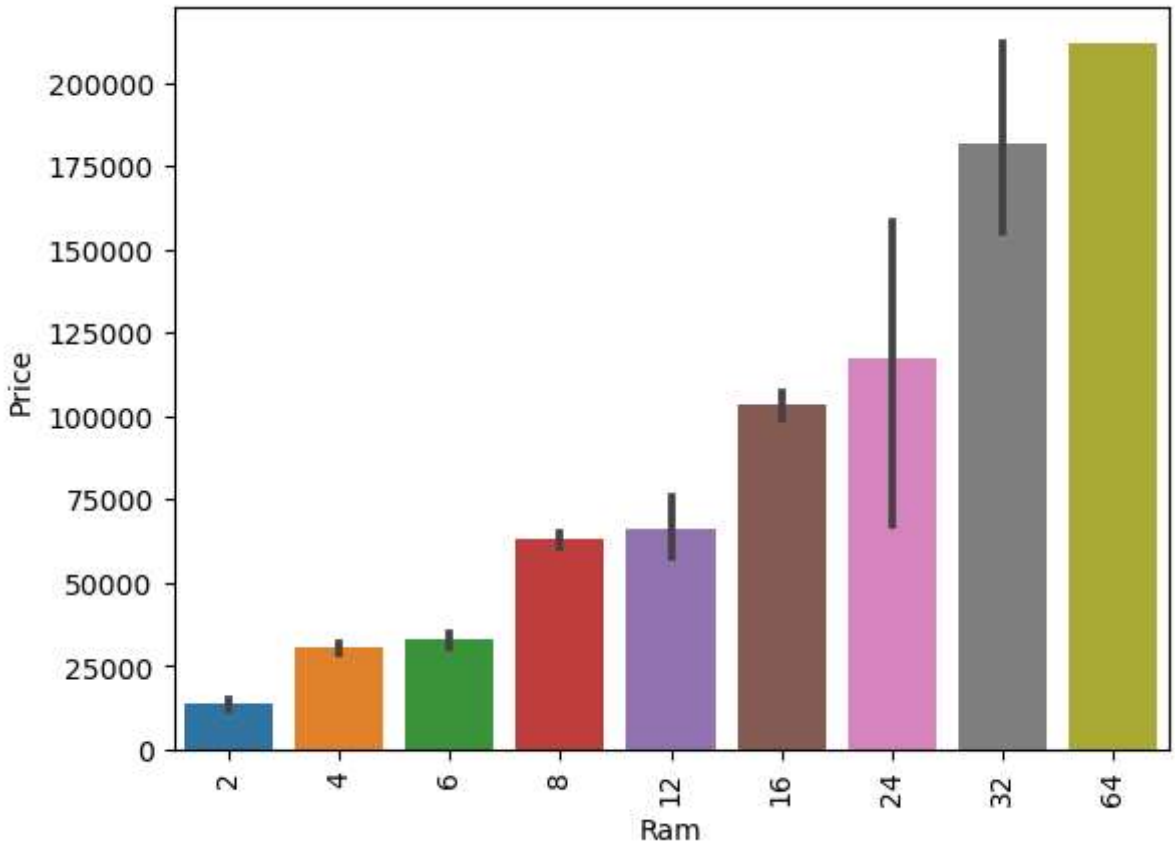
Out[338]:

```
<Axes: >
```



In [339…

```
sns.barplot(x=df['Ram'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



In [344…

```
df['Memory'].value_counts()
```

Out[344]:

```
256 SSD                           412
1000 HDD                          224
500 HDD                           132
512 SSD                           118
128 SSD +  1000 HDD                94
128 SSD                            76
256 SSD +  1000 HDD                73
32 Flash Storage                   38
2000 HDD                           16
64 Flash Storage                   15
512 SSD +  1000 HDD                14
1000 SSD                           14
256 SSD +  2000 HDD                10
1000 Hybrid                         9
256 Flash Storage                   8
16 Flash Storage                    7
32 SSD                              6
180 SSD                             5
128 Flash Storage                   4
512 SSD +  2000 HDD                 3
16 SSD                              3
512 Flash Storage                   2
1000 SSD +  1000 HDD                2
256 SSD +  500 HDD                  2
128 SSD +  2000 HDD                 2
256 SSD +  256 SSD                  2
512 SSD +  256 SSD                  1
512 SSD +  512 SSD                  1
64 Flash Storage +  1000 HDD        1
1000 HDD +  1000 HDD                1
32 HDD                              1
64 SSD                              1
128 HDD                             1
240 SSD                             1
8 SSD                               1
508 Hybrid                          1
512 SSD +  1000 Hybrid              1
256 SSD +  1000 Hybrid              1
Name: Memory, dtype: int64
```

In [345…

```
df['Memory'] = df['Memory'].astype(str).replace('\.0', '', regex=True)
df["Memory"] = df["Memory"].str.replace('GB', '')
df["Memory"] = df["Memory"].str.replace('TB', '000')
new = df["Memory"].str.split("+", n = 1, expand = True)

df["first"]= new[0]
df["first"]=df["first"].str.strip()

df["second"]= new[1]

df["Layer1HDD"] = df["first"].apply(lambda x: 1 if "HDD" in x else 0)
df["Layer1SSD"] = df["first"].apply(lambda x: 1 if "SSD" in x else 0)
df["Layer1Hybrid"] = df["first"].apply(lambda x: 1 if "Hybrid" in x else 0)
df["Layer1Flash_Storage"] = df["first"].apply(lambda x: 1 if "Flash Storage" in x else

df['first'] = df['first'].str.replace(r'\D', '',regex=True)

df["second"].fillna("0", inplace = True)

df["Layer2HDD"] = df["second"].apply(lambda x: 1 if "HDD" in x else 0)
```

```
df["Layer2SSD"] = df["second"].apply(lambda x: 1 if "SSD" in x else 0)
df["Layer2Hybrid"] = df["second"].apply(lambda x: 1 if "Hybrid" in x else 0)
df["Layer2Flash_Storage"] = df["second"].apply(lambda x: 1 if "Flash Storage" in x els

df['second'] = df['second'].str.replace(r'\D', '',regex=True)

df["first"] = df["first"].astype(int)
df["second"] = df["second"].astype(int)

df["HDD"]=(df["first"]*df["Layer1HDD"]+df["second"]*df["Layer2HDD"])
df["SSD"]=(df["first"]*df["Layer1SSD"]+df["second"]*df["Layer2SSD"])
df["Hybrid"]=(df["first"]*df["Layer1Hybrid"]+df["second"]*df["Layer2Hybrid"])
df["Flash_Storage"]=(df["first"]*df["Layer1Flash_Storage"]+df["second"]*df["Layer2Flas

df.drop(columns=['first', 'second', 'Layer1HDD', 'Layer1SSD', 'Layer1Hybrid',
       'Layer1Flash_Storage', 'Layer2HDD', 'Layer2SSD', 'Layer2Hybrid',
       'Layer2Flash_Storage'],inplace=True)
```

In [346…

```
df.sample(5)
```

Out[346]:

	Company	TypeName	Ram	Memory	Gpu	OpSys	Weight	Price	Touchscreen	Ip
118	Asus	Notebook	4	1000 HDD	Nvidia GeForce 920MX	Windows 10	2.00	30049.920	0	
696	Lenovo	2 in 1 Convertible	16	512 SSD	Intel HD Graphics 620	Windows 10	1.38	93186.720	1	
20	Asus	Netbook	2	32 Flash Storage	Intel HD Graphics 400	Windows 10	0.98	10224.432	0	
473	Dell	Ultrabook	8	256 SSD	Intel UHD Graphics 620	Windows 10	1.42	98133.768	1	
1159	HP	2 in 1 Convertible	8	512 SSD	Intel HD Graphics 520	Windows 10	1.48	108744.480	1	

In [347…

```
df.drop(columns=['Memory'],inplace=True)
```

In [348…

```
df.head()
```

Out[348]:

	Company	TypeName	Ram	Gpu	OpSys	Weight	Price	Touchscreen	Ips	ppi
0	Apple	Ultrabook	8	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	226.983005
1	Apple	Ultrabook	8	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	127.677940
2	HP	Notebook	8	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	141.211998
3	Apple	Ultrabook	16	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	220.534624
4	Apple	Ultrabook	8	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1	226.983005

In [349…
```
df.corr(numeric_only = True)['Price']
```

Out[349]:
```
Ram              0.743007
Weight           0.210370
Price            1.000000
Touchscreen      0.191226
Ips              0.252208
ppi              0.473487
HDD             -0.096441
SSD              0.670799
Hybrid           0.007989
Flash_Storage   -0.040511
Name: Price, dtype: float64
```

In [350…
```
df.drop(columns=['Hybrid','Flash_Storage'],inplace=True)
```

In [351…
```
df.head()
```

Out[351]:

	Company	TypeName	Ram	Gpu	OpSys	Weight	Price	Touchscreen	Ips	ppi
0	Apple	Ultrabook	8	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	226.983005
1	Apple	Ultrabook	8	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	127.677940
2	HP	Notebook	8	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	141.211998
3	Apple	Ultrabook	16	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	220.534624
4	Apple	Ultrabook	8	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1	226.983005

In [352…

```
df['Gpu'].value_counts()
```

Out[352]:

```
Intel HD Graphics 620      281
Intel HD Graphics 520      185
Intel UHD Graphics 620      68
Nvidia GeForce GTX 1050     66
Nvidia GeForce GTX 1060     48
                          ...
AMD Radeon R5 520            1
AMD Radeon R7                1
Intel HD Graphics 540        1
AMD Radeon 540               1
ARM Mali T860 MP4            1
Name: Gpu, Length: 110, dtype: int64
```

In [353…

```
df['Gpu brand'] = df['Gpu'].apply(lambda x:x.split()[0])
```

In [354…

```
df.head()
```

Out[354]:

	Company	TypeName	Ram	Gpu	OpSys	Weight	Price	Touchscreen	Ips	ppi
0	Apple	Ultrabook	8	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	226.983005
1	Apple	Ultrabook	8	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	127.677940
2	HP	Notebook	8	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	141.211998
3	Apple	Ultrabook	16	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	220.534624
4	Apple	Ultrabook	8	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1	226.983005

In [355…
```
df['Gpu brand'].value_counts()
```

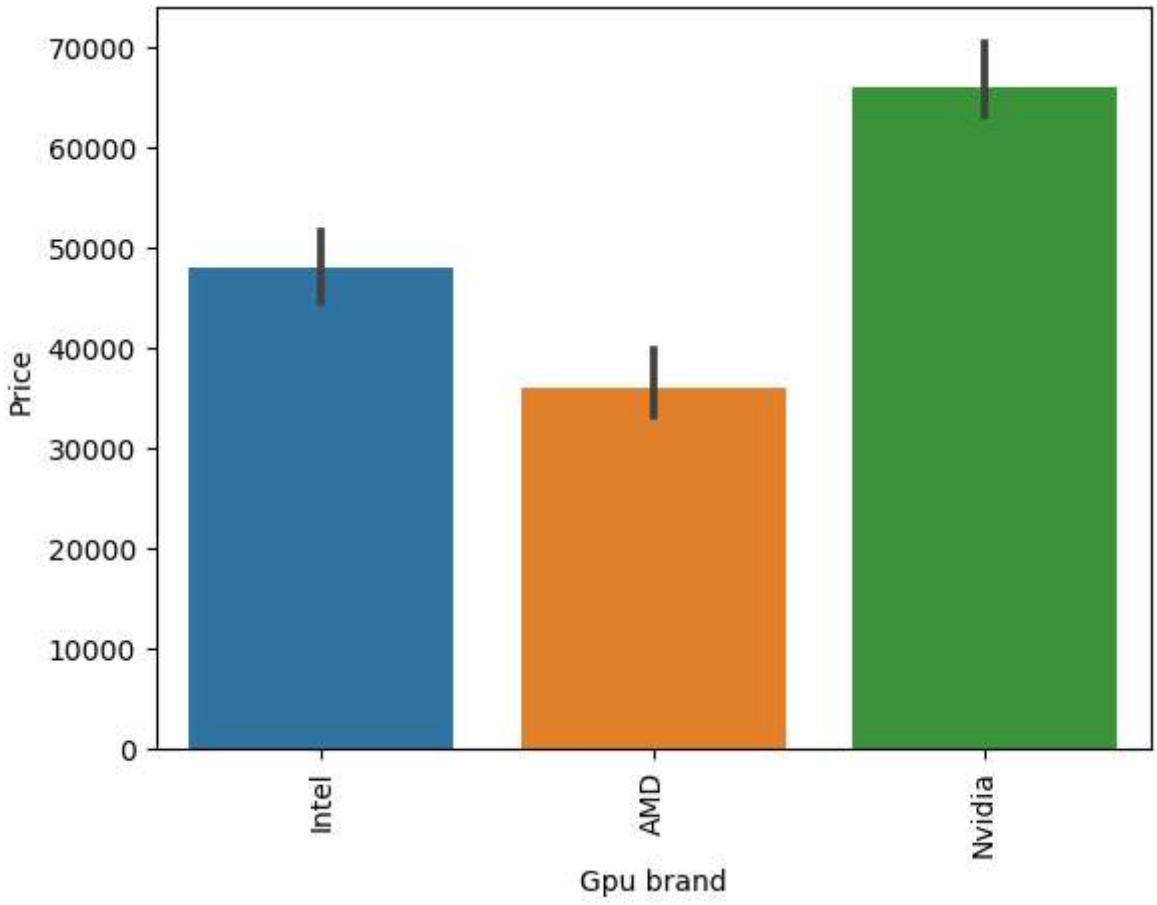
Out[355]:
```
Intel     722
Nvidia    400
AMD       180
ARM         1
Name: Gpu brand, dtype: int64
```

In [356…
```
df = df[df['Gpu brand'] != 'ARM']
```

In [357…
```
df['Gpu brand'].value_counts()
```

Out[357]:
```
Intel     722
Nvidia    400
AMD       180
Name: Gpu brand, dtype: int64
```

In [358…
```
sns.barplot(x=df['Gpu brand'],y=df['Price'],estimator=np.median)
plt.xticks(rotation='vertical')
plt.show()
```



In [359…
```
df.drop(columns=['Gpu'],inplace=True)
```

In [360…
```
df.head()
```

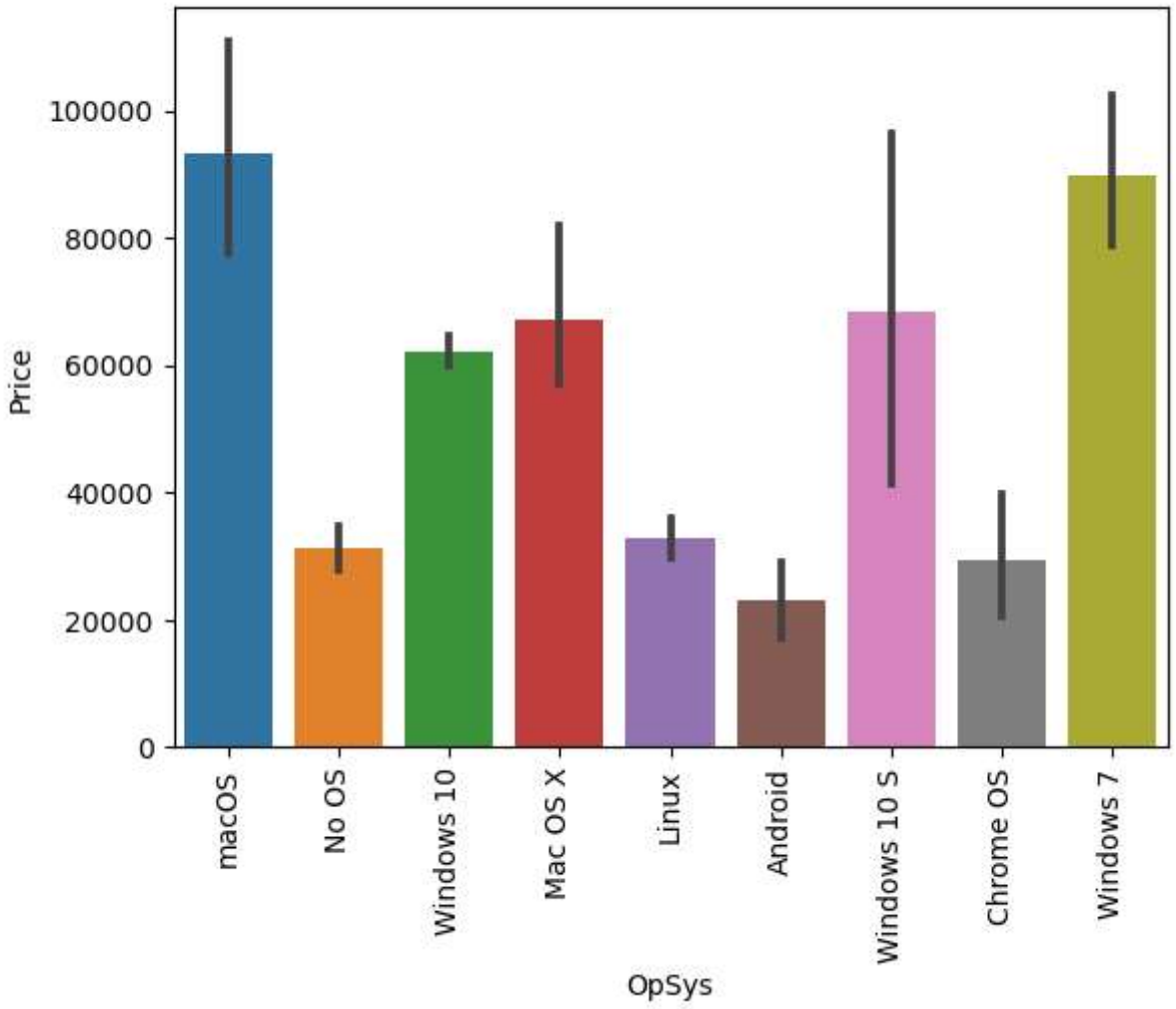
Out[360]:

	Company	TypeName	Ram	OpSys	Weight	Price	Touchscreen	Ips	ppi	Cpu brand	H
0	Apple	Ultrabook	8	macOS	1.37	71378.6832	0	1	226.983005	Intel Core i5	
1	Apple	Ultrabook	8	macOS	1.34	47895.5232	0	0	127.677940	Intel Core i5	
2	HP	Notebook	8	No OS	1.86	30636.0000	0	0	141.211998	Intel Core i5	
3	Apple	Ultrabook	16	macOS	1.83	135195.3360	0	1	220.534624	Intel Core i7	
4	Apple	Ultrabook	8	macOS	1.37	96095.8080	0	1	226.983005	Intel Core i5	

In [361…
```
df['OpSys'].value_counts()
```

Out[361]:
```
Windows 10      1072
No OS             66
Linux             62
Windows 7         45
Chrome OS         26
macOS             13
Mac OS X           8
Windows 10 S       8
Android            2
Name: OpSys, dtype: int64
```

In [362…
```
sns.barplot(x=df['OpSys'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



In [363…
```
def cat_os(inp):
    if inp == 'Windows 10' or inp == 'Windows 7' or inp == 'Windows 10 S':
        return 'Windows'
    elif inp == 'macOS' or inp == 'Mac OS X':
        return 'Mac'
    else:
        return 'Others/No OS/Linux'
```

In [364…
```
df['os'] = df['OpSys'].apply(cat_os)
```

In [365…
```
df.head()
```

Out[365]:

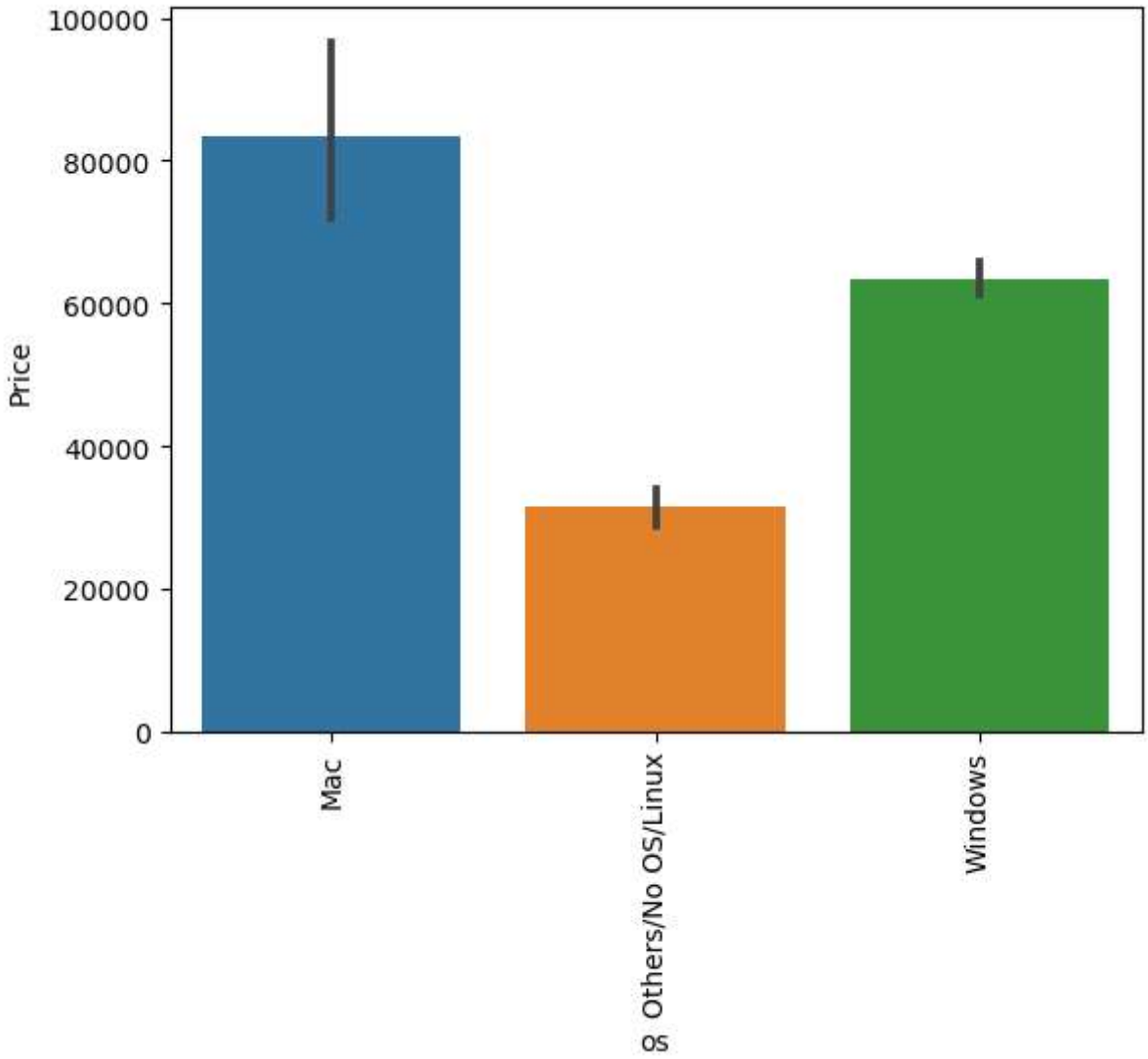
	Company	TypeName	Ram	OpSys	Weight	Price	Touchscreen	Ips	ppi	Cpu brand	H
0	Apple	Ultrabook	8	macOS	1.37	71378.6832	0	1	226.983005	Intel Core i5	
1	Apple	Ultrabook	8	macOS	1.34	47895.5232	0	0	127.677940	Intel Core i5	
2	HP	Notebook	8	No OS	1.86	30636.0000	0	0	141.211998	Intel Core i5	
3	Apple	Ultrabook	16	macOS	1.83	135195.3360	0	1	220.534624	Intel Core i7	
4	Apple	Ultrabook	8	macOS	1.37	96095.8080	0	1	226.983005	Intel Core i5	

In [366…

```
df.drop(columns=['OpSys'],inplace=True)
```

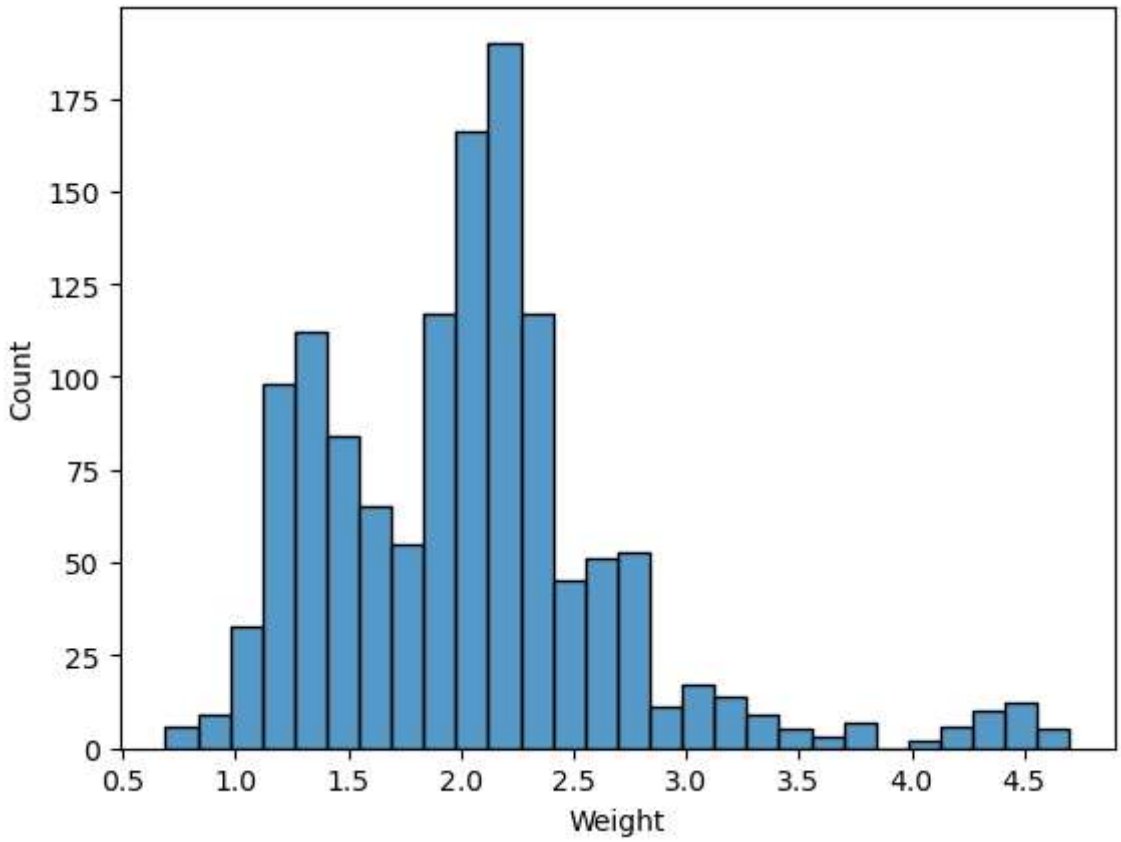
In [367…

```
sns.barplot(x=df['os'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



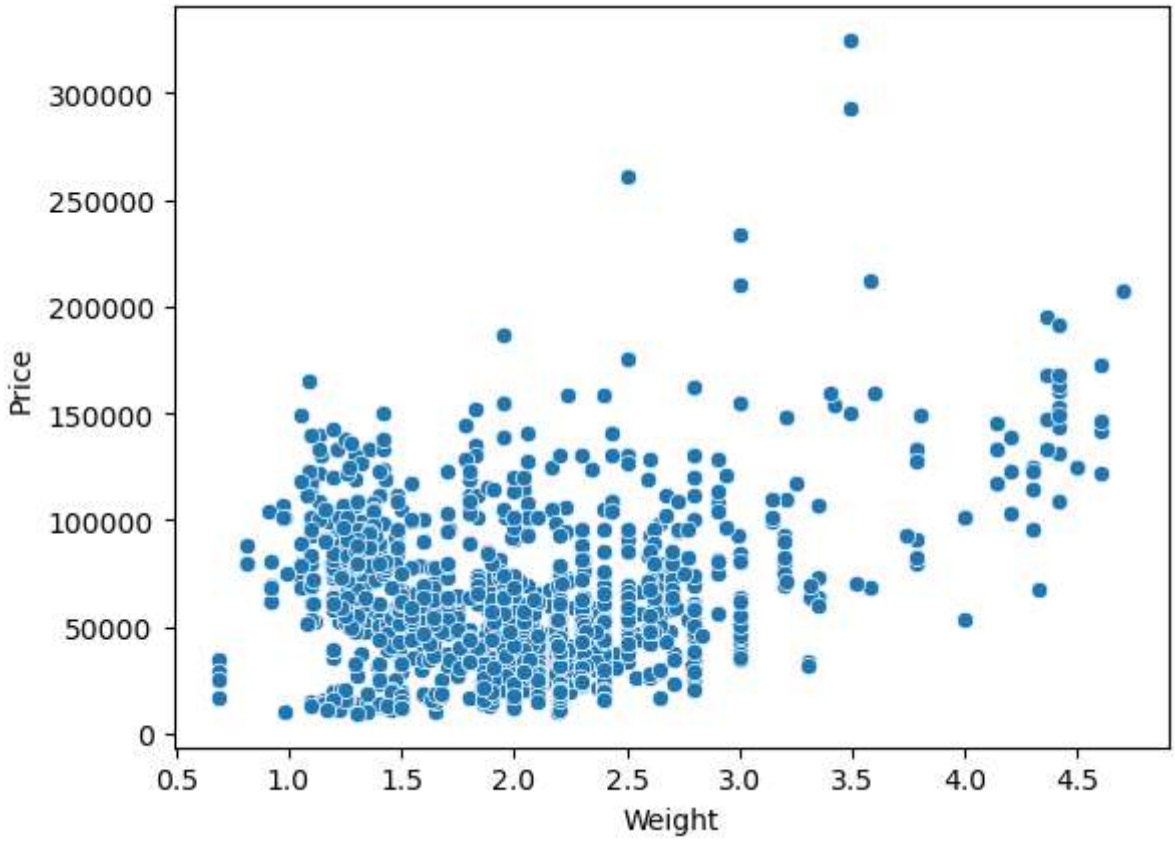
In [368…
```
sns.histplot(df['Weight'])
```

Out[368]:
```
<Axes: xlabel='Weight', ylabel='Count'>
```



In [369…
```
sns.scatterplot(x=df['Weight'],y=df['Price'])
```

Out[369]:
```
<Axes: xlabel='Weight', ylabel='Price'>
```

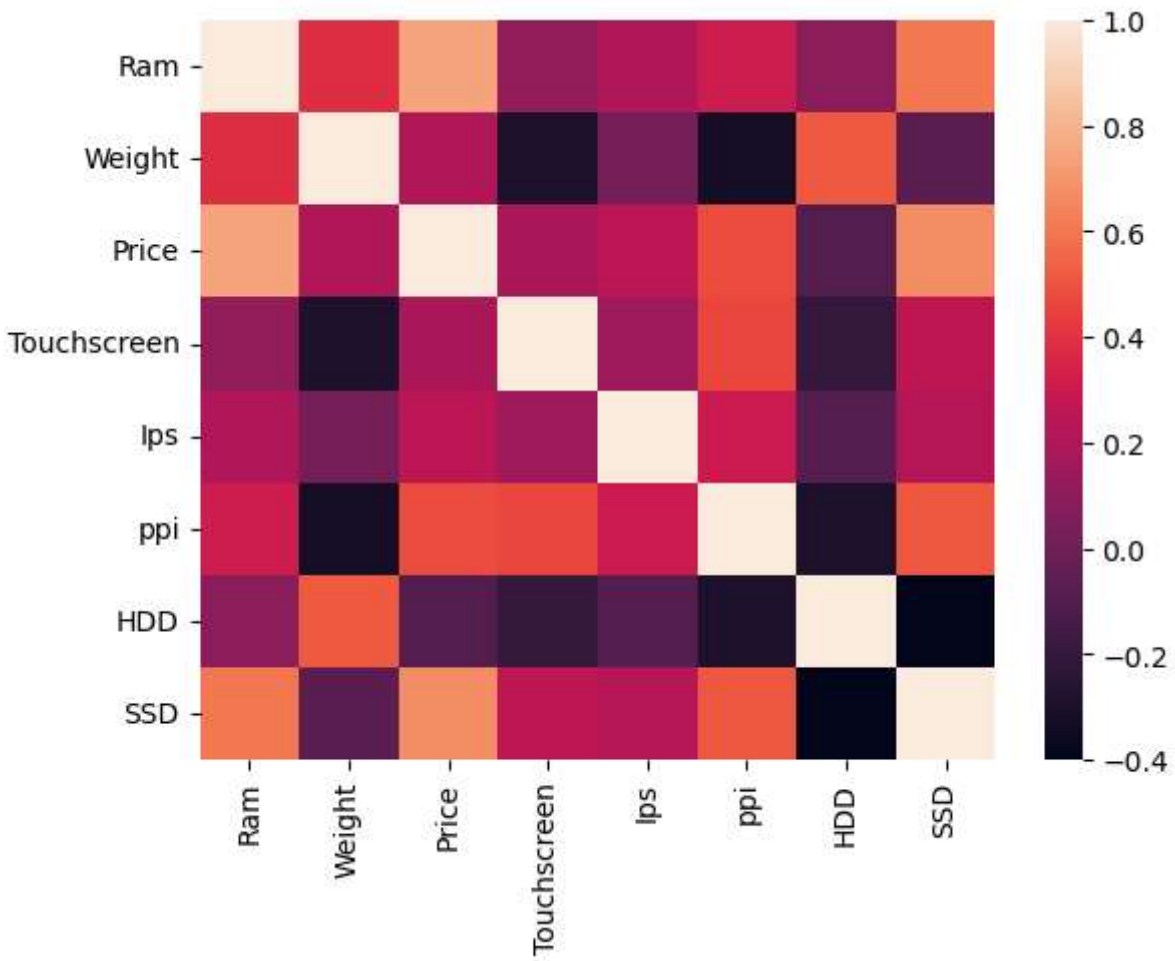


In [370…
```
df.corr(numeric_only = True)['Price']
```

Out[370]:
```
Ram            0.742905
Weight         0.209867
Price          1.000000
Touchscreen    0.192917
Ips            0.253320
ppi            0.475368
HDD           -0.096891
SSD            0.670660
Name: Price, dtype: float64
```

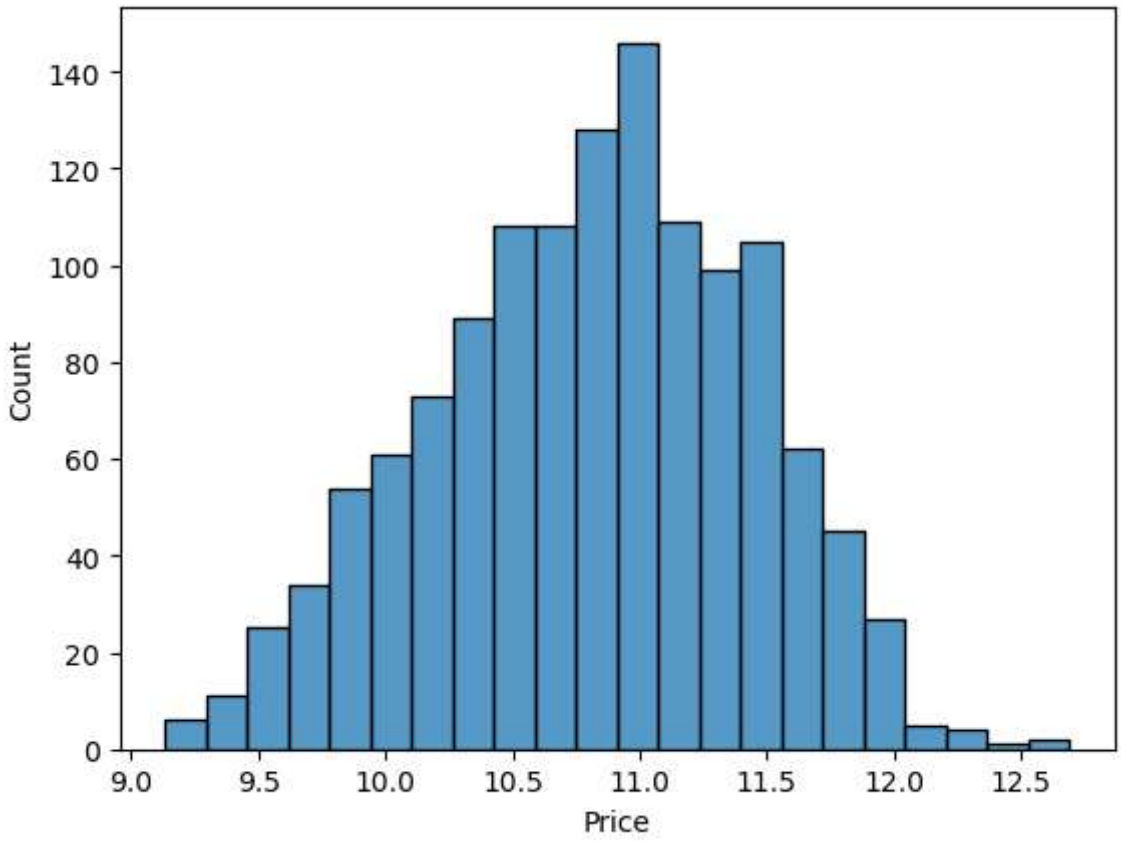
In [371…
```
sns.heatmap(df.corr(numeric_only = True))
```

Out[371]:
```
<Axes: >
```



In [372…
```
sns.histplot(np.log(df['Price']))
```

Out[372]:
```
<Axes: xlabel='Price', ylabel='Count'>
```



In [373…
```
X = df.drop(columns=['Price'])
y = np.log(df['Price'])
```

In [374…
```
X
```

Out[374]:

	Company	TypeName	Ram	Weight	Touchscreen	Ips	ppi	Cpu brand	HDD	SSD	Gp bran
0	Apple	Ultrabook	8	1.37	0	1	226.983005	Intel Core i5	0	128	Int
1	Apple	Ultrabook	8	1.34	0	0	127.677940	Intel Core i5	0	0	Int
2	HP	Notebook	8	1.86	0	0	141.211998	Intel Core i5	0	256	Int
3	Apple	Ultrabook	16	1.83	0	1	220.534624	Intel Core i7	0	512	AM
4	Apple	Ultrabook	8	1.37	0	1	226.983005	Intel Core i5	0	256	Int
...	...	...	...	...	...	...	...	...	...	...	
1298	Lenovo	2 in 1 Convertible	4	1.80	1	1	157.350512	Intel Core i7	0	128	Int
1299	Lenovo	2 in 1 Convertible	16	1.30	1	1	276.053530	Intel Core i7	0	512	Int
1300	Lenovo	Notebook	2	1.50	0	0	111.935204	Other Intel Processor	0	0	Int
1301	HP	Notebook	6	2.19	0	0	100.454670	Intel Core i7	1000	0	AM
1302	Asus	Notebook	4	2.20	0	0	100.454670	Other Intel Processor	500	0	Int

1302 rows × 12 columns

In [375…
```
y
```

Out[375]:
```
0       11.175755
1       10.776777
2       10.329931
3       11.814476
4       11.473101
          ...
1298    10.433899
1299    11.288115
1300     9.409283
1301    10.614129
1302     9.886358
Name: Price, Length: 1302, dtype: float64
```

In [376…
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.15,random_state=2)
```

In [377…
```
X_train
```

Out[377]:

	Company	TypeName	Ram	Weight	Touchscreen	Ips	ppi	Cpu brand	HDD	SSD	Gp bran
183	Toshiba	Notebook	8	2.00	0	0	100.454670	Intel Core i5	0	128	Int
1141	MSI	Gaming	8	2.40	0	0	141.211998	Intel Core i7	1000	128	Nvid
1049	Asus	Netbook	4	1.20	0	0	135.094211	Other Intel Processor	0	0	Int
1020	Dell	2 in 1 Convertible	4	2.08	1	1	141.211998	Intel Core i3	1000	0	Int
878	Dell	Notebook	4	2.18	0	0	141.211998	Intel Core i5	1000	128	Nvid
...	...	...	...	...	...	...	...	...	...	...	
466	Acer	Notebook	4	2.20	0	0	100.454670	Intel Core i3	500	0	Nvid
299	Asus	Ultrabook	16	1.63	0	0	141.211998	Intel Core i7	0	512	Nvid
493	Acer	Notebook	8	2.20	0	0	100.454670	AMD Processor	1000	0	AM
527	Lenovo	Notebook	8	2.20	0	0	100.454670	Intel Core i3	2000	0	Nvid
1193	Apple	Ultrabook	8	0.92	0	1	226.415547	Other Intel Processor	0	0	Int

1106 rows × 12 columns

In [384...
```
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import r2_score,mean_absolute_error
```

In [385...
```
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor, AdaBoos
from sklearn.svm import SVR
from xgboost import XGBRegressor
```

Linear regression

In [386...
```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = LinearRegression()
```

```
pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.8073277448418712
MAE 0.2101782797642824
```

Ridge Regression

In [387…

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = Ridge(alpha=10)

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.8127331031311801
MAE 0.20926802242583054
```

Lasso Regression

In [388…

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = Lasso(alpha=0.001)

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.8071853945317105
MAE 0.21114361613472565
```

KNN

In [389…

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = KNeighborsRegressor(n_neighbors=3)

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.803148868705085
MAE 0.19264883332948865
```

Decision Tree

In [390…

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = DecisionTreeRegressor(max_depth=8)

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.8400517762196216
MAE 0.1841289093756122
```

SVM

In [392…

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = SVR(kernel='rbf',C=10000,epsilon=0.1)
```

```
pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.808318090228966
MAE 0.20239059427193437
```

Random Forest

In [393…

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = RandomForestRegressor(n_estimators=100,
                              random_state=3,
                              max_samples=0.5,
                              max_features=0.75,
                              max_depth=15)

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.8875592075383176
MAE 0.15876547703102858
```

ExtraTrees

In [396…

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = ExtraTreesRegressor(n_estimators=100,
                              random_state=3,
                              max_samples=None,
                              max_features=0.75,
                              max_depth=15)

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])
```

```
pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.8743576019640713
MAE 0.1606945068551793
```

AdaBoost

In [398...

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = AdaBoostRegressor(n_estimators=15,learning_rate=1.0)

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.7807822437662586
MAE 0.23833274140582608
```

Gradient Boost

In [399...

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = GradientBoostingRegressor(n_estimators=500)

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.881634191365353
MAE 0.15993118561599914
```

XgBoost

In [400…

```
step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = XGBRegressor(n_estimators=45,max_depth=5,learning_rate=0.5)

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.8811773435850243
MAE 0.16496203512600974
```

Voting Regressor

In [404…

```
from sklearn.ensemble import VotingRegressor,StackingRegressor

step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')


rf = RandomForestRegressor(n_estimators=350,random_state=3,max_samples=None,max_featur
gbdt = GradientBoostingRegressor(n_estimators=100,max_features=0.5)
xgb = XGBRegressor(n_estimators=25,learning_rate=0.3,max_depth=5)
et = ExtraTreesRegressor(n_estimators=100,random_state=3,max_samples=None,max_features

step2 = VotingRegressor([('rf', rf), ('gbdt', gbdt), ('xgb',xgb), ('et',et)],weights=

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.8916030877069989
MAE 0.15639365963808996
```

Stacking

In [405…

```
from sklearn.ensemble import VotingRegressor,StackingRegressor

step1 = ColumnTransformer(transformers=[
```

```
    ('col_tnf',OneHotEncoder(sparse_output=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')


estimators = [
    ('rf', RandomForestRegressor(n_estimators=350,random_state=3,max_samples=0.5,max_f
    ('gbdt',GradientBoostingRegressor(n_estimators=100,max_features=0.5)),
    ('xgb', XGBRegressor(n_estimators=25,learning_rate=0.3,max_depth=5))
]

step2 = StackingRegressor(estimators=estimators, final_estimator=Ridge(alpha=100))

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

```
R2 score 0.8792759669675067
MAE 0.16787541507916198
```

Exporting the Model

In [406…

```
import pickle

pickle.dump(df,open('df.pkl','wb'))
pickle.dump(pipe,open('pipe.pkl','wb'))
```

In [407…

```
df
```

Out[407]:

	Company	TypeName	Ram	Weight	Price	Touchscreen	Ips	ppi	Cpu brand	HD
0	Apple	Ultrabook	8	1.37	71378.6832	0	1	226.983005	Intel Core i5	
1	Apple	Ultrabook	8	1.34	47895.5232	0	0	127.677940	Intel Core i5	
2	HP	Notebook	8	1.86	30636.0000	0	0	141.211998	Intel Core i5	
3	Apple	Ultrabook	16	1.83	135195.3360	0	1	220.534624	Intel Core i7	
4	Apple	Ultrabook	8	1.37	96095.8080	0	1	226.983005	Intel Core i5	
...	...	...	...	...	...	...	...	...	...	
1298	Lenovo	2 in 1 Convertible	4	1.80	33992.6400	1	1	157.350512	Intel Core i7	
1299	Lenovo	2 in 1 Convertible	16	1.30	79866.7200	1	1	276.053530	Intel Core i7	
1300	Lenovo	Notebook	2	1.50	12201.1200	0	0	111.935204	Other Intel Processor	
1301	HP	Notebook	6	2.19	40705.9200	0	0	100.454670	Intel Core i7	100
1302	Asus	Notebook	4	2.20	19660.3200	0	0	100.454670	Other Intel Processor	50

1302 rows × 13 columns

In [408…

```
X_train
```

Out[408]:

	Company	TypeName	Ram	Weight	Touchscreen	Ips	ppi	Cpu brand	HDD	SSD	Gp bran
183	Toshiba	Notebook	8	2.00	0	0	100.454670	Intel Core i5	0	128	Int
1141	MSI	Gaming	8	2.40	0	0	141.211998	Intel Core i7	1000	128	Nvid
1049	Asus	Netbook	4	1.20	0	0	135.094211	Other Intel Processor	0	0	Int
1020	Dell	2 in 1 Convertible	4	2.08	1	1	141.211998	Intel Core i3	1000	0	Int
878	Dell	Notebook	4	2.18	0	0	141.211998	Intel Core i5	1000	128	Nvid
...	...	...	...	...	...	...	...	...	...	...	
466	Acer	Notebook	4	2.20	0	0	100.454670	Intel Core i3	500	0	Nvid
299	Asus	Ultrabook	16	1.63	0	0	141.211998	Intel Core i7	0	512	Nvid
493	Acer	Notebook	8	2.20	0	0	100.454670	AMD Processor	1000	0	AM
527	Lenovo	Notebook	8	2.20	0	0	100.454670	Intel Core i3	2000	0	Nvid
1193	Apple	Ultrabook	8	0.92	0	1	226.415547	Other Intel Processor	0	0	Int

1106 rows × 12 columns

In [ ]:

In [ ]:

In [ ]: