In [276…
```
import numpy as np
import pandas as pd
```

In [277…
```
df=pd.read_csv('laptop_data.csv')
```

In [278…
```
df.head()
```

Out[278…

	Unnamed: 0	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	
0	0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	In Gr
1	1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	In Gr
2	2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	In Gr
3	3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	R P
4	4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	In Gr

In [279…
```
df.shape
```

Out[279…
```
(1303, 12)
```

In [280…
```
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 [281…
```
df.duplicated().sum()
```

Out[281…
```
0
```

In [282…
```
df.isnull().sum()
```

Out[282…
```
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 [283…
```
df.drop(columns=['Unnamed: 0'],inplace=True)
```

In [284…
```
df.head()
```

Out[284...

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

In [285...

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

In [286...

```
df.head()
```

Out[286…

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

In [287…
```
df['Ram']=df['Ram'].astype('int32')
df['Weight']=df['Weight'].astype('float32')
```

In [288…
```
df.info()
```

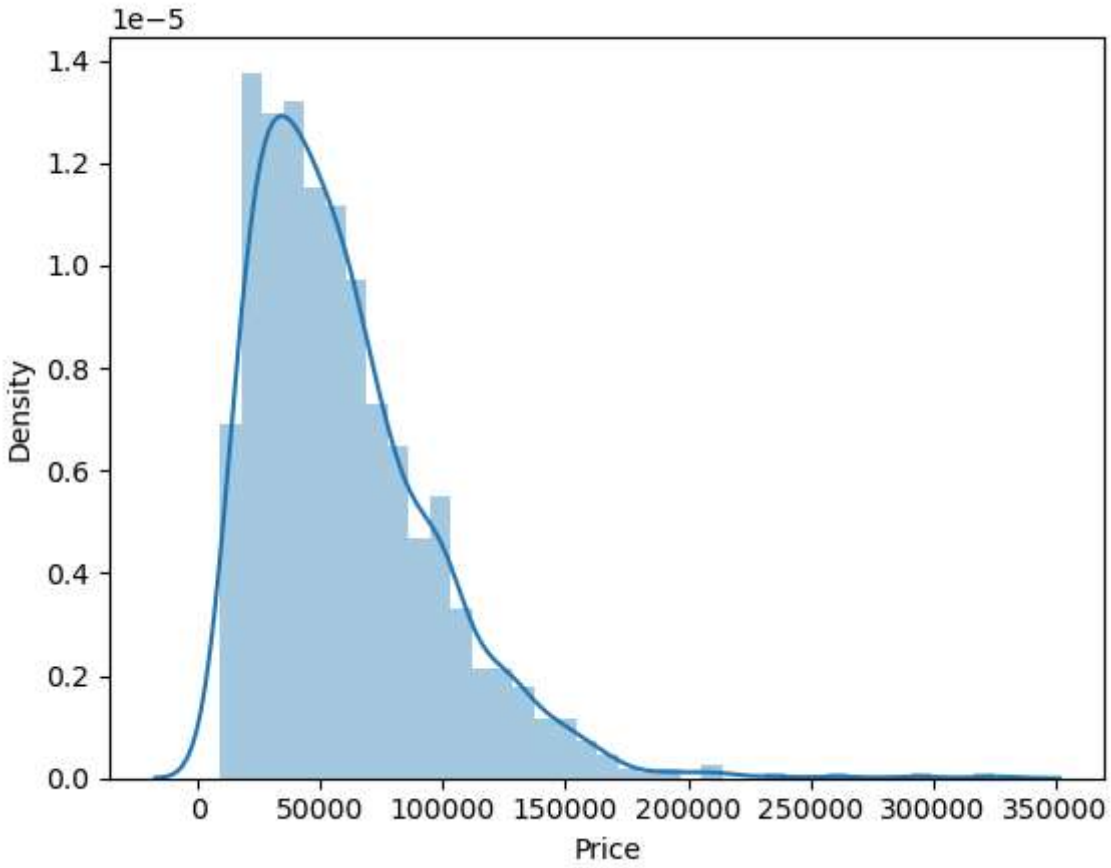
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Company           1303 non-null   object 
 1   TypeName          1303 non-null   object 
 2   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 [289…
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [290…
```
sns.distplot(df['Price'])
```

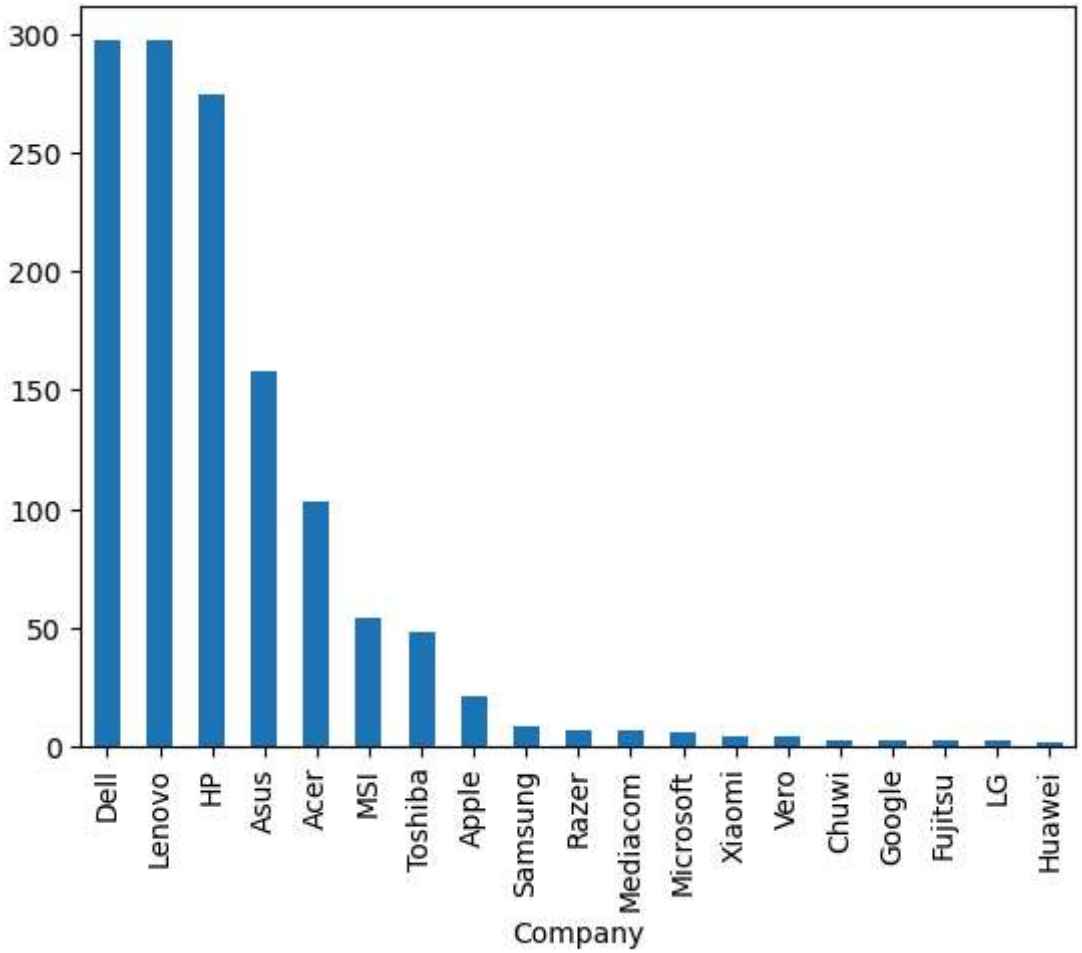
```
C:\Users\adity\AppData\Local\Temp\ipykernel_10220\834922981.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Price'])
```

Out[290… `<Axes: xlabel='Price', ylabel='Density'>`



In [291…
```
df['Company'].value_counts().plot(kind='bar')
```

Out[291… `<Axes: xlabel='Company'>`

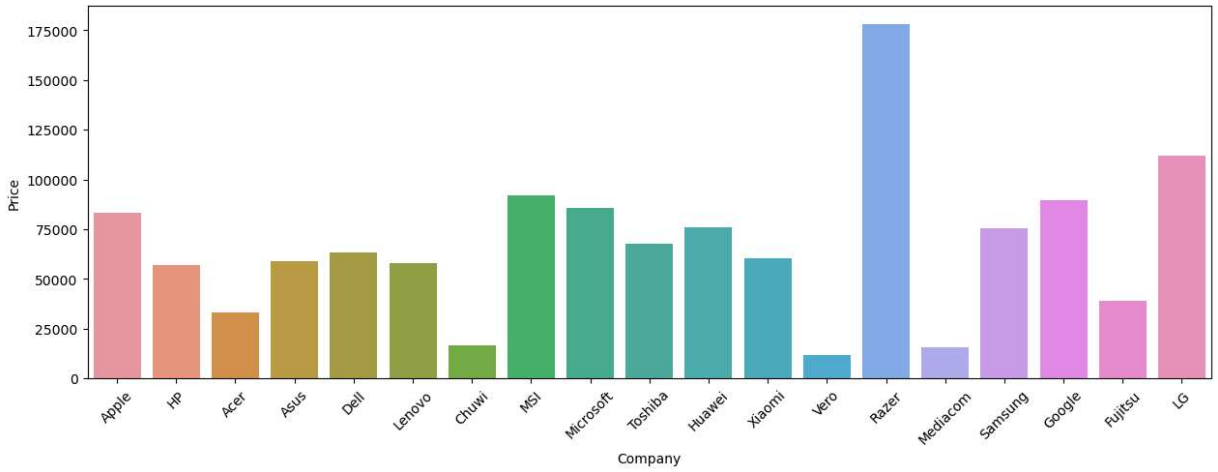


In [292…

```
plt.figure(figsize=(15,5))
sns.barplot(data=df,x='Company',y='Price',ci=False)
plt.xticks(rotation=45)
```

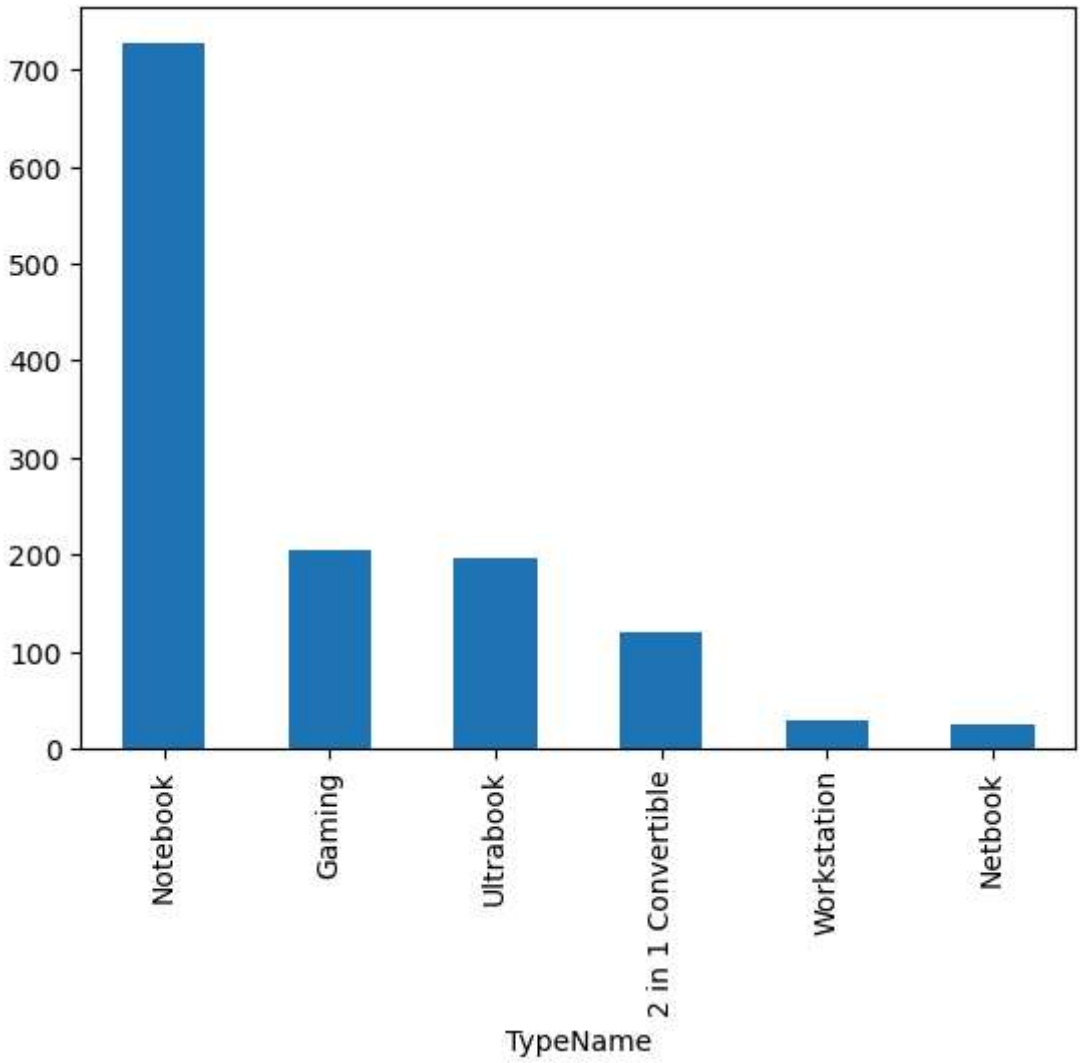
```
C:\Users\adity\AppData\Local\Temp\ipykernel_10220\1880596878.py:2: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=('ci', False)` for the same effect.

  sns.barplot(data=df,x='Company',y='Price',ci=False)
```

Out[292…
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18]),
 [Text(0, 0, 'Apple'),
  Text(1, 0, 'HP'),
  Text(2, 0, 'Acer'),
  Text(3, 0, 'Asus'),
  Text(4, 0, 'Dell'),
  Text(5, 0, 'Lenovo'),
  Text(6, 0, 'Chuwi'),
  Text(7, 0, 'MSI'),
  Text(8, 0, 'Microsoft'),
  Text(9, 0, 'Toshiba'),
  Text(10, 0, 'Huawei'),
  Text(11, 0, 'Xiaomi'),
  Text(12, 0, 'Vero'),
  Text(13, 0, 'Razer'),
  Text(14, 0, 'Mediacom'),
  Text(15, 0, 'Samsung'),
  Text(16, 0, 'Google'),
  Text(17, 0, 'Fujitsu'),
  Text(18, 0, 'LG')])
```



In [293…
```
df['TypeName'].value_counts().plot(kind='bar')
```

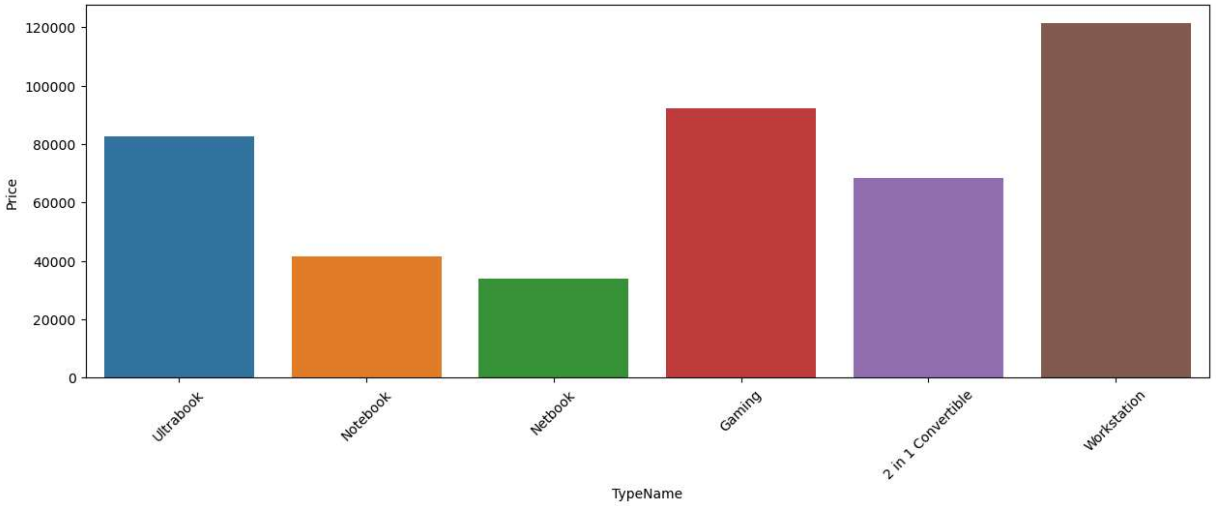
Out[293…
```
<Axes: xlabel='TypeName'>
```



In [294…
```
plt.figure(figsize=(15,5))
sns.barplot(data=df,x='TypeName',y='Price',ci=False)
plt.xticks(rotation=45)
```

```
C:\Users\adity\AppData\Local\Temp\ipykernel_10220\2979882505.py:2: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=('ci', False)` for the same effect.

  sns.barplot(data=df,x='TypeName',y='Price',ci=False)
```

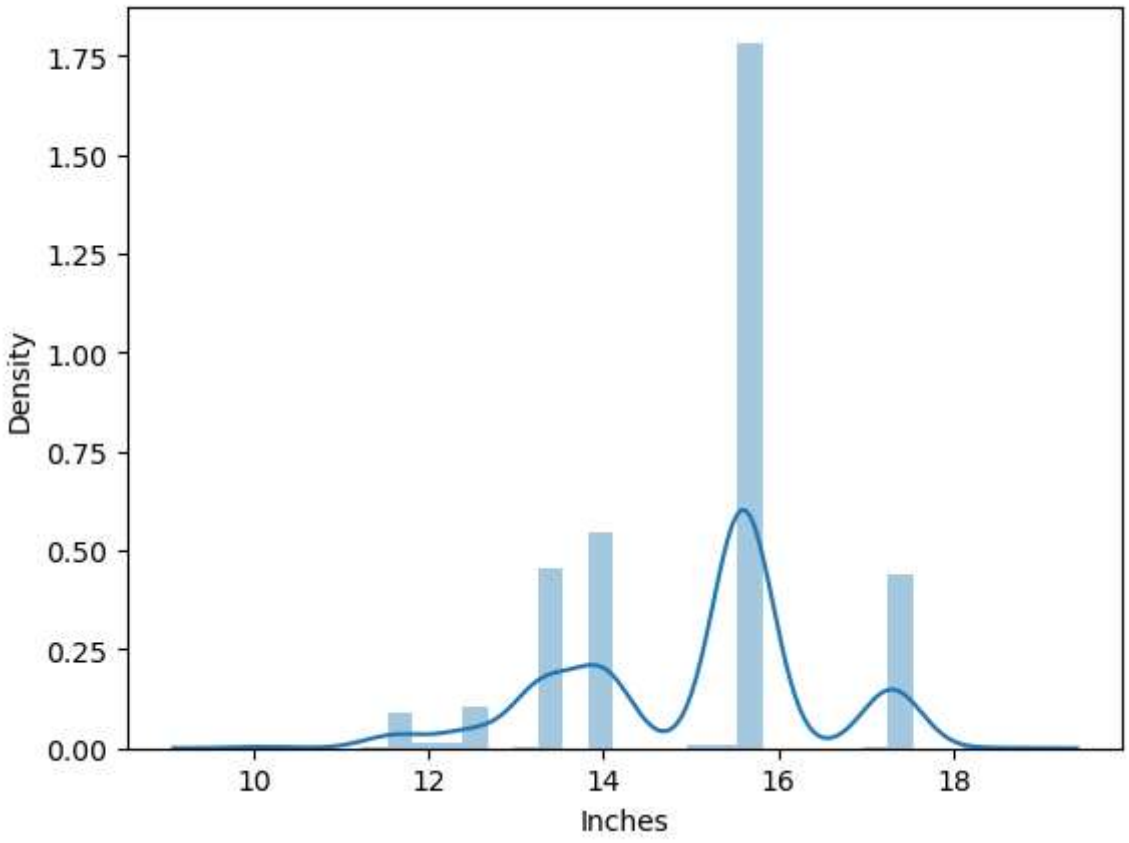
Out[294…
```
(array([0, 1, 2, 3, 4, 5]),
 [Text(0, 0, 'Ultrabook'),
  Text(1, 0, 'Notebook'),
  Text(2, 0, 'Netbook'),
  Text(3, 0, 'Gaming'),
  Text(4, 0, '2 in 1 Convertible'),
  Text(5, 0, 'Workstation')])
```



In [295…
```
sns.distplot(df['Inches'])
```

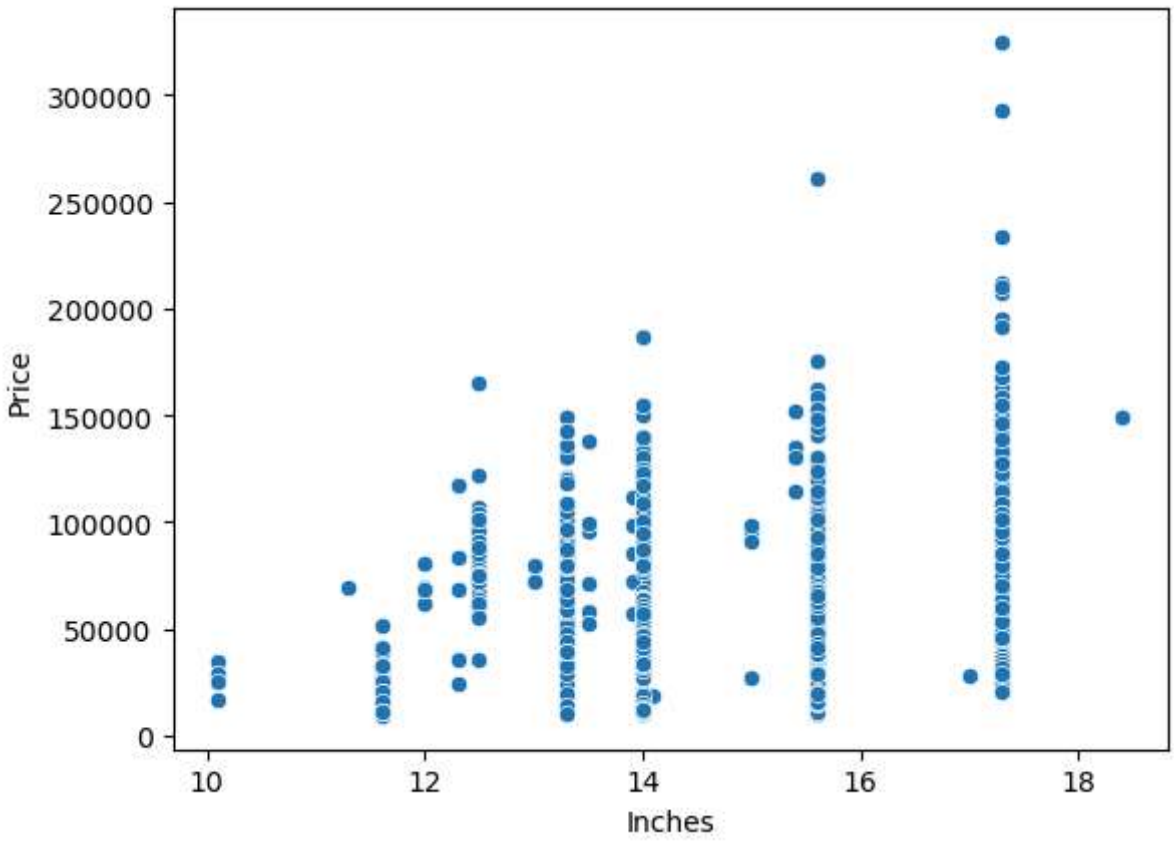
```
C:\Users\adity\AppData\Local\Temp\ipykernel_10220\1439577752.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Inches'])
```

Out[295…
```
<Axes: xlabel='Inches', ylabel='Density'>
```



In [296…
```
sns.scatterplot(x=df['Inches'],y=df['Price'])
```

Out[296…
```
<Axes: xlabel='Inches', ylabel='Price'>
```



In [297…
```
df['ScreenResolution'].value_counts()
```

Out[297…
```
ScreenResolution
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: count, dtype: int64
```

In [298…
```
df['Touchscreen']=df['ScreenResolution'].apply(lambda x:1 if 'Touchscreen' in x els
```

In [299…
```
df.sample(5)
```

Out[299…

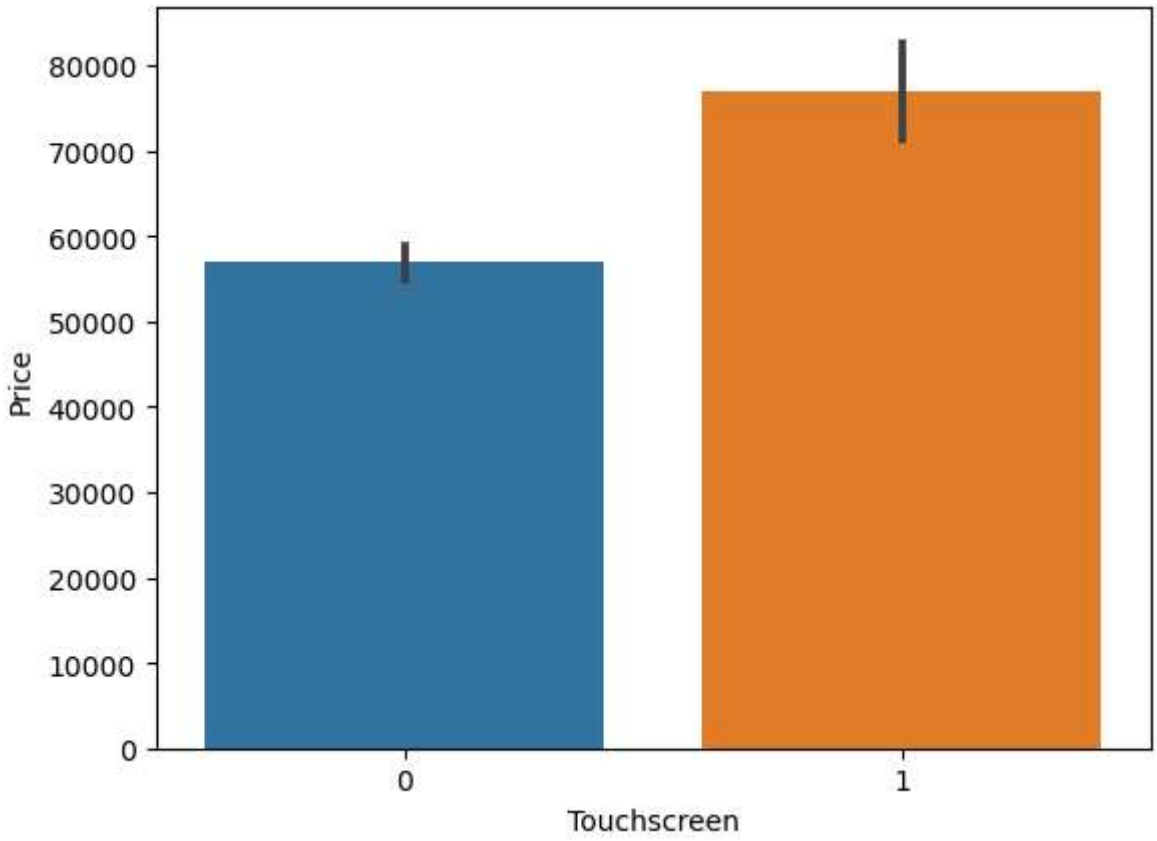
	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu
624	HP	Notebook	15.6	1366x768	Intel Core i5 6200U 2.3GHz	4	500GB HDD	Intel HD Graphics 520
1160	Lenovo	Ultrabook	14.0	Full HD 1920x1080	Intel Core i7 6600U 2.6GHz	8	256GB SSD	Intel HD Graphics 520
1228	Lenovo	Gaming	15.6	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i7 6700HQ 2.6GHz	16	128GB SSD + 1TB HDD	Nvidia GeForce GTX 960M
1046	HP	Notebook	14.0	Full HD 1920x1080	Intel Core i5 6200U 2.3GHz	8	256GB SSD	Intel HD Graphics 520
17	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.9GHz	16	512GB SSD	AMD Radeon Pro 560

In [300…
```
df['Touchscreen'].value_counts()
```

Out[300…
```
Touchscreen
0    1111
1     192
Name: count, dtype: int64
```

In [301…
```
sns.barplot(x=df['Touchscreen'],y=df['Price'])
```

Out[301…
```
<Axes: xlabel='Touchscreen', ylabel='Price'>
```



In [302…
```
df['IPS']=df['ScreenResolution'].apply(lambda x:1 if 'IPS' in x else 0)
```

In [303…
```
df.sample(5)
```

Out[303…

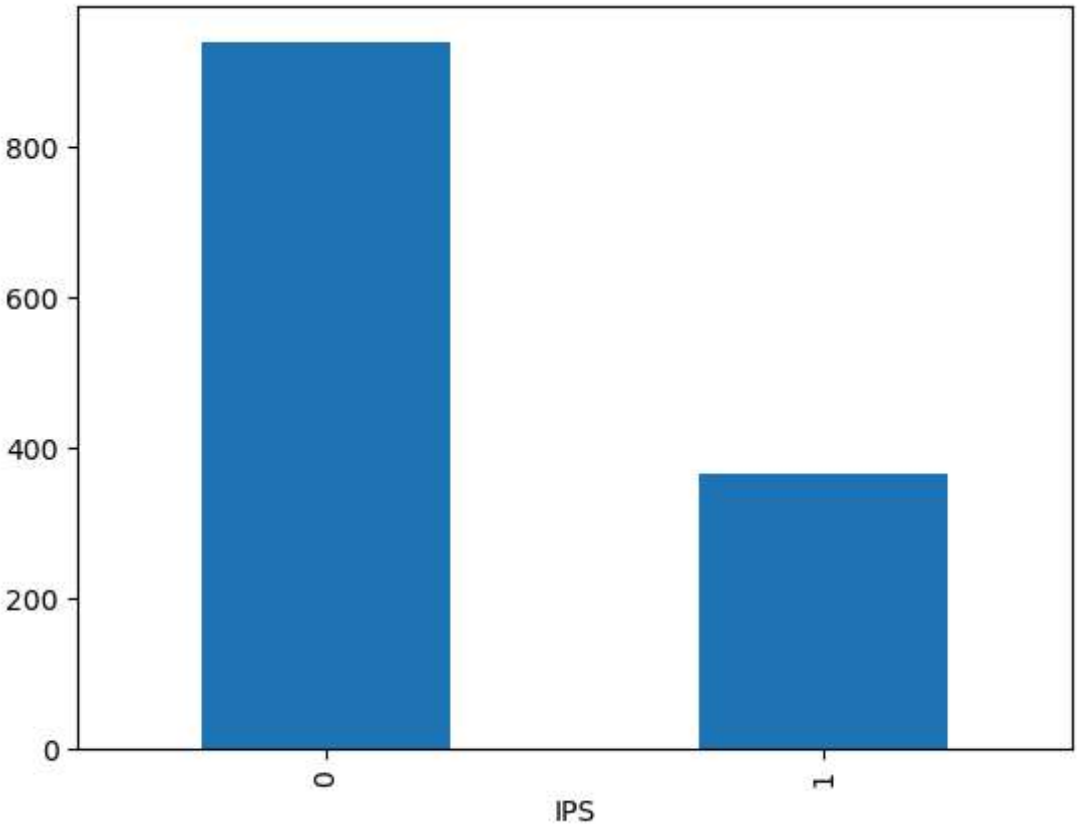
	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	
916	HP	Notebook	13.3	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	W
895	Toshiba	Notebook	13.3	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	W
1060	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i7 7500U 2.7GHz	8	256GB SSD	Intel HD Graphics 620	W
751	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 6300U 2.4GHz	8	256GB SSD	Intel HD Graphics 520	W
1078	Lenovo	Notebook	15.6	1366x768	Intel Core i5 6200U 2.3GHz	4	500GB HDD	Intel HD Graphics 520	W

In [304…

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

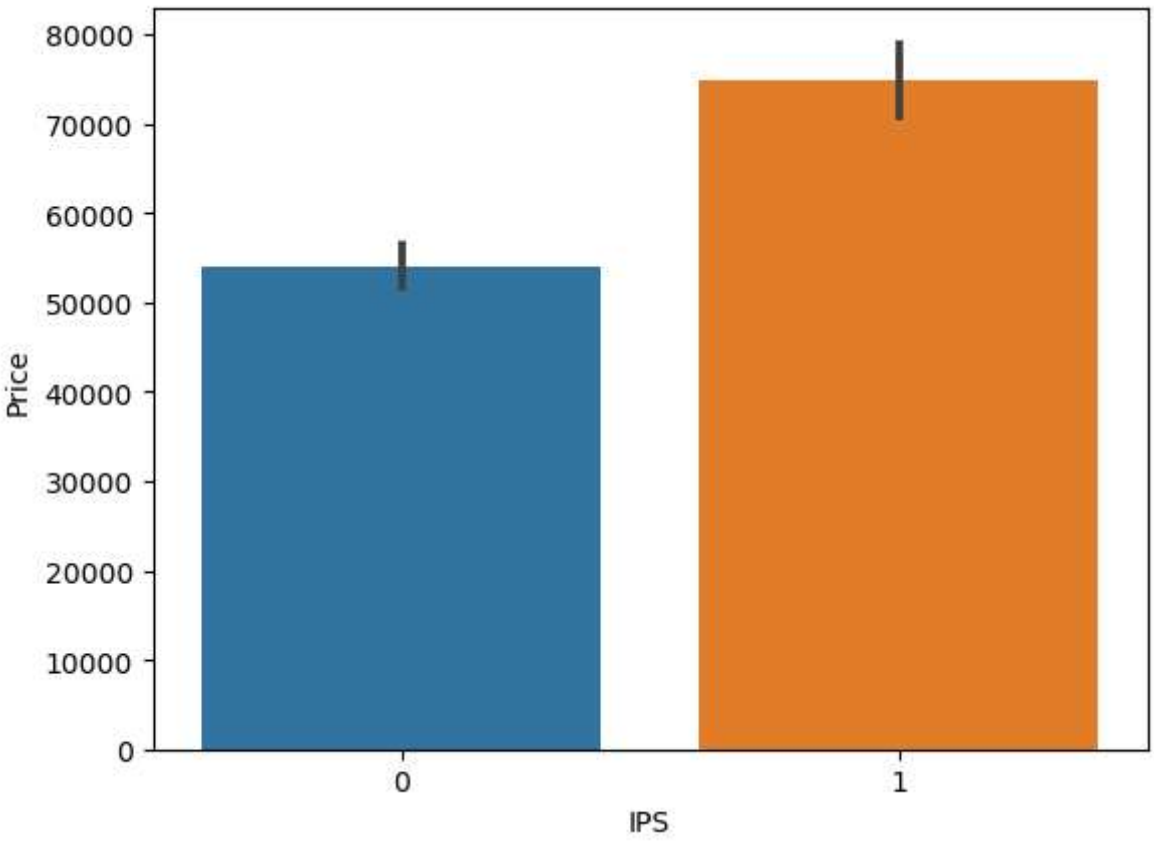
Out[304…

```
<Axes: xlabel='IPS'>
```



In [305…
```
sns.barplot(data=df, x='IPS', y='Price')
```

Out[305…
```
<Axes: xlabel='IPS', ylabel='Price'>
```



In [306…
```
new=df['ScreenResolution'].str.split('x',expand=True)
```

In [307…
```
df['x_res']=new[0]
df['y_res']=new[1]
```

In [308…
```
df.head()
```

Out[308…

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

In [309…
```
df['x_res'] = df['x_res'].str.split(" ").str.get(-1)
```

In [310…
```
df.head()
```

Out[310…

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

In [311…

```
df['x_res']=df['x_res'].astype('int')
df['y_res']=df['y_res'].astype('int')
```

In [312…

```
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 [313…

```
df.select_dtypes(include=['int','float']).corr()['Price']
```

Out[313…
```
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 [314…
```
df['ppi']=((df['x_res']**2+df['y_res']**2))**0.5/df['Inches'].astype('float')
```

In [315…
```
df.select_dtypes(include=['int','float']).corr()['Price']
```

Out[315…
```
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 [316…
```
df.drop(columns=['ScreenResolution'],inplace=True)
```

In [317…
```
df.head()
```

Out[317…

	Company	TypeName	Inches	Cpu	Ram	Memory	Gpu	OpSys	Weight	
0	Apple	Ultrabook	13.3	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378
1	Apple	Ultrabook	13.3	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895
2	HP	Notebook	15.6	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636
3	Apple	Ultrabook	15.4	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195
4	Apple	Ultrabook	13.3	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095

In [318…
```
df.drop(columns=['Inches','x_res','y_res'],inplace=True)
```

In [319…
```
df.head()
```

Out[319…

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

In [320…
```
df['Cpu'].value_counts()
```

Out[320…
```
Cpu
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: count, Length: 118, dtype: int64
```

In [321…
```
df['Cpu_name']=df['Cpu'].apply(lambda x:" ".join(x.split()[0:3]))
```

In [322…
```
df.head()
```

Out[322...

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

In [323...

```
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 [324...

```
df['Cpu brand']= df['Cpu_name'].apply(fetch_processor)
```

In [325...

```
df.head()
```

Out[325…

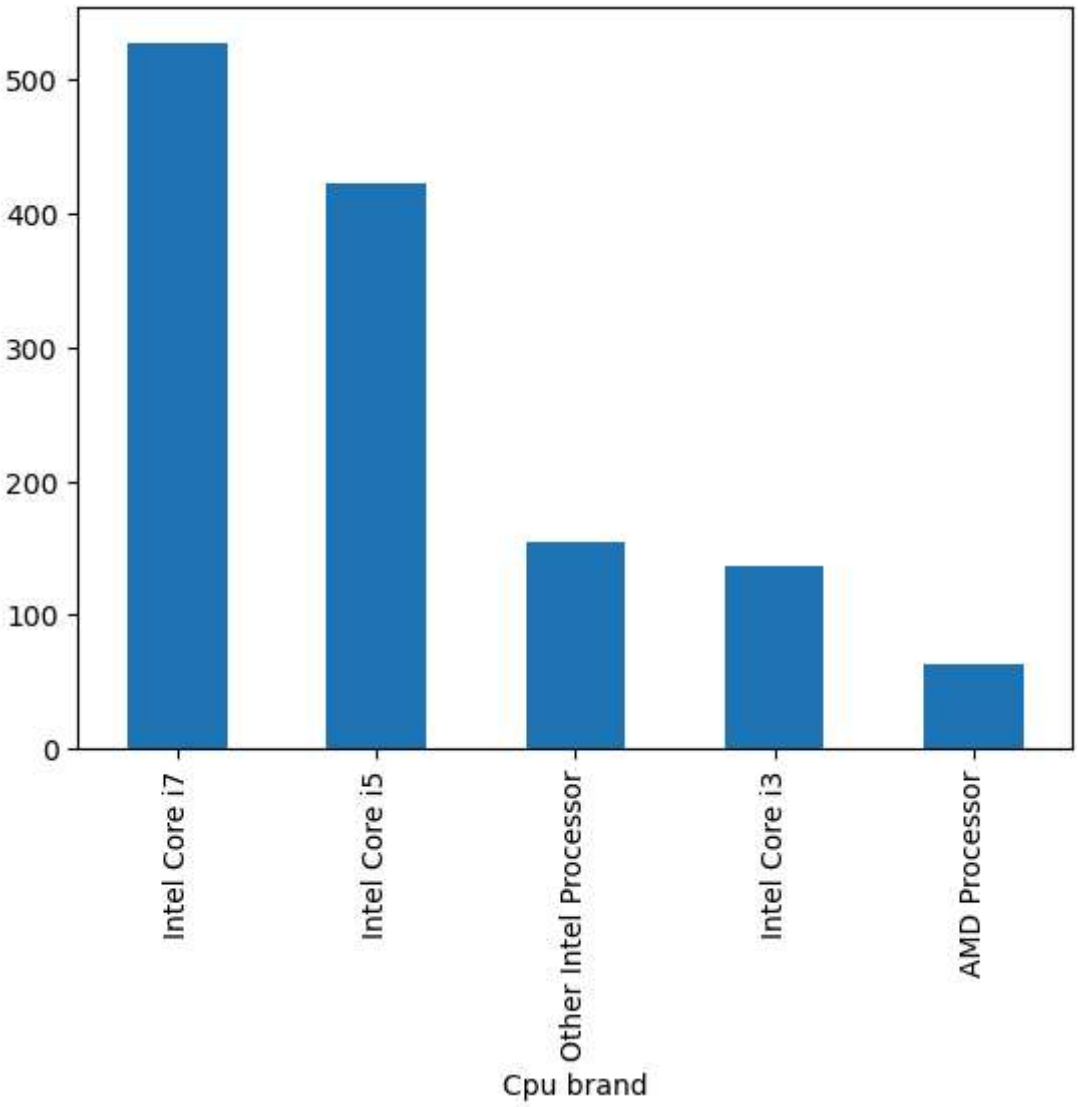
	Company	TypeName	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	T
0	Apple	Ultrabook	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	
1	Apple	Ultrabook	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	
2	HP	Notebook	Intel Core i5 7200U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

In [326…

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

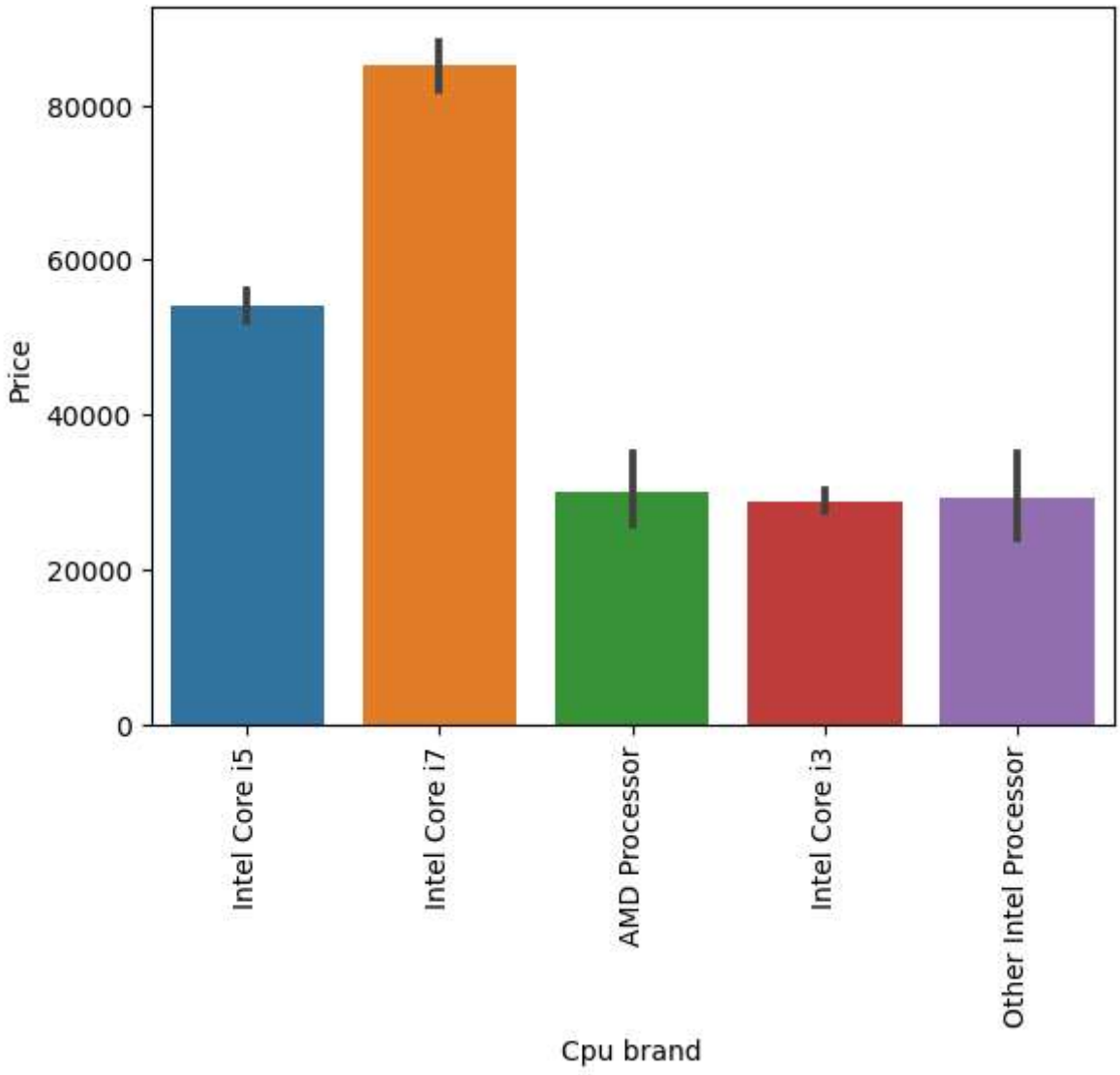
Out[326…

```
<Axes: xlabel='Cpu brand'>
```



In [327…
```
sns.barplot(x=df['Cpu brand'],y=df['Price'])
plt.xticks(rotation=90)
```

Out[327…
```
(array([0, 1, 2, 3, 4]),
 [Text(0, 0, 'Intel Core i5'),
  Text(1, 0, 'Intel Core i7'),
  Text(2, 0, 'AMD Processor'),
  Text(3, 0, 'Intel Core i3'),
  Text(4, 0, 'Other Intel Processor')])
```



In [328…
```
df.drop(columns=['Cpu','Cpu_name'],inplace=True)
```

In [329…
```
df.head()
```

Out[329…

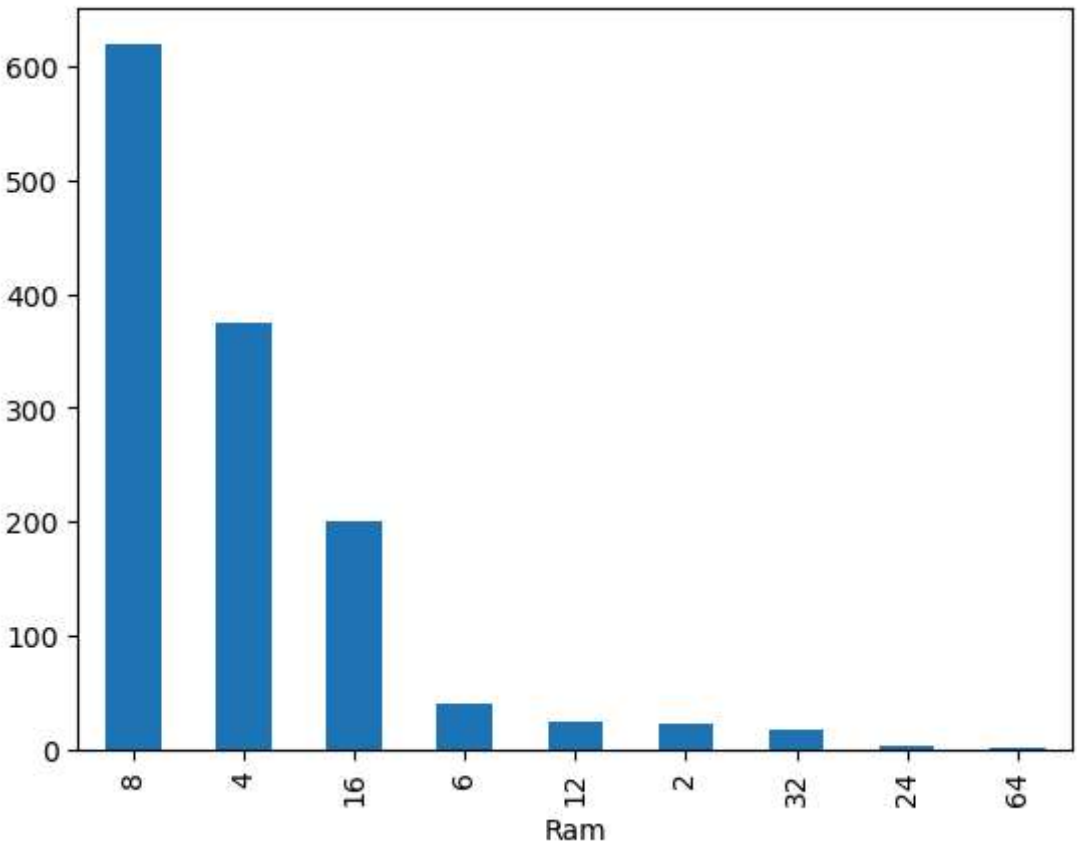
	Company	TypeName	Ram	Memory	Gpu	OpSys	Weight	Price	Touchscre
0	Apple	Ultrabook	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	
1	Apple	Ultrabook	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	
2	HP	Notebook	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	
3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

In [330…

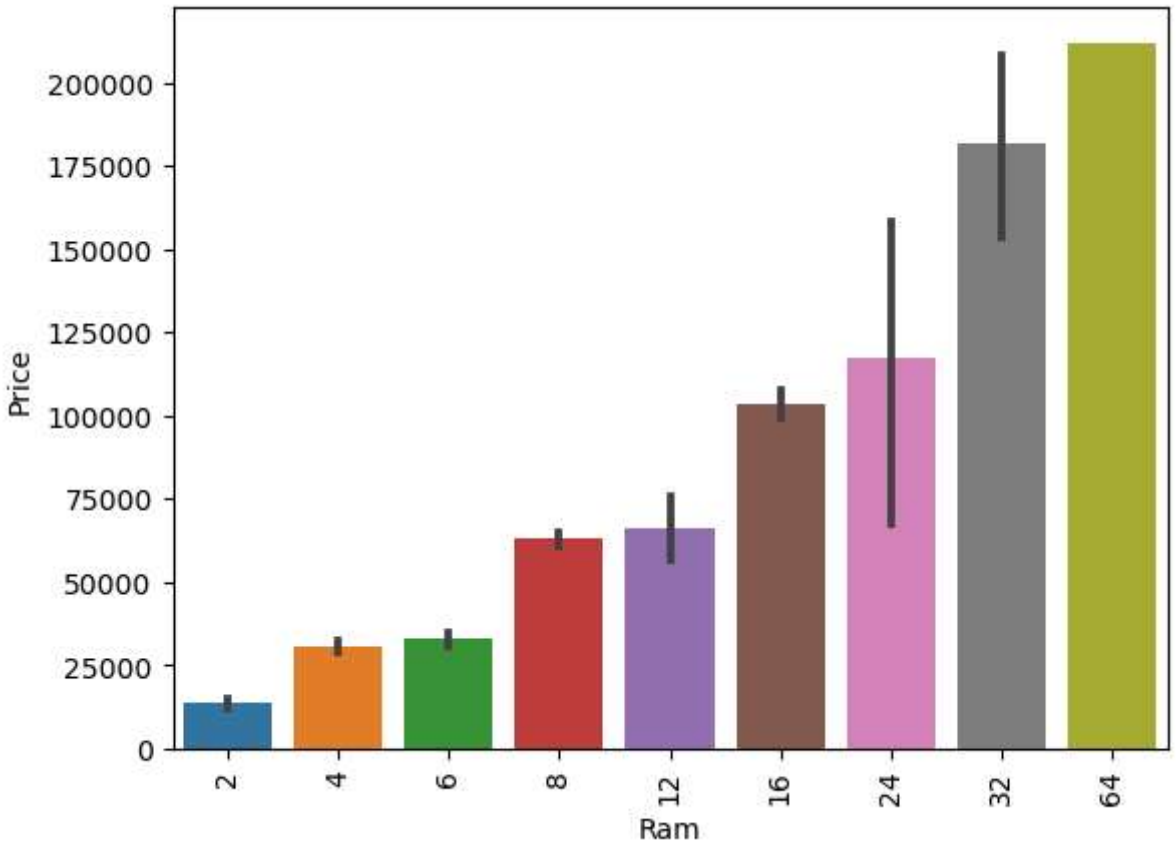
```
df['Ram'].value_counts().plot(kind='bar')
```

Out[330…

```
<Axes: xlabel='Ram'>
```



In [331…
```
sns.barplot(x=df['Ram'],y=df['Price'])
plt.xticks(rotation=90)
plt.show()
```



In [332…
```
df['Memory'].value_counts()
```

Out[332…

```
Memory
256GB SSD                        412
1TB HDD                          223
500GB HDD                        132
512GB SSD                        118
128GB SSD +  1TB HDD              94
128GB SSD                         76
256GB SSD +  1TB HDD              73
32GB Flash Storage                38
2TB HDD                           16
64GB Flash Storage                15
512GB SSD +  1TB HDD              14
1TB SSD                           14
256GB SSD +  2TB HDD              10
1.0TB Hybrid                       9
256GB Flash Storage                8
16GB Flash Storage                 7
32GB SSD                           6
180GB SSD                          5
128GB Flash Storage                4
512GB SSD +  2TB HDD               3
16GB SSD                           3
512GB Flash Storage                2
1TB SSD +  1TB HDD                 2
256GB SSD +  500GB HDD             2
128GB SSD +  2TB HDD               2
256GB SSD +  256GB SSD             2
512GB SSD +  256GB SSD             1
512GB SSD +  512GB SSD             1
64GB Flash Storage +  1TB HDD      1
1TB HDD +  1TB HDD                 1
32GB HDD                           1
64GB SSD                           1
128GB HDD                          1
240GB SSD                          1
8GB SSD                            1
508GB Hybrid                       1
1.0TB HDD                          1
512GB SSD +  1.0TB Hybrid          1
256GB SSD +  1.0TB Hybrid          1
Name: count, dtype: int64
```

In [333…

```
# Remove decimals and units from memory
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')

# Split memory configurations into layers
new = df['Memory'].str.split('+', n=1, expand=True)
df['first'] = new[0].str.strip()
df['second'] = new[1].fillna('0').str.strip()

# Identify storage types in each layer
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 e

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

# Extract numeric values
df['first'] = df['first'].str.extract(r'(\d+)').astype(float)
df['first'] = df['first'].fillna(0).astype(int)

df['second'] = df['second'].str.extract(r'(\d+)').astype(float)
df['second'] = df['second'].fillna(0).astype(int)


# Calculate total storage for each type
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['

# Drop redundant columns
df.drop(columns=['first', 'second', 'Layer1SSD', 'Layer1HDD', 'Layer1Hybrid', 'Laye
                 'Layer2SSD', 'Layer2HDD', 'Layer2Hybrid', 'Layer2Flash Storage'],
```

In [334…

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

Out[334…

	Company	TypeName	Ram	Memory	Gpu	OpSys	Weight	Price	Touchs
217	HP	Notebook	8	256 SSD	Nvidia GeForce 930MX	Windows 10	1.63	54931.68	
1021	Toshiba	Ultrabook	8	256 SSD	Intel HD Graphics 520	Windows 10	1.20	84715.20	
580	Mediacom	Notebook	4	32 SSD	Intel HD Graphics 500	Windows 10	1.45	20725.92	
870	Lenovo	Notebook	4	500 HDD	Intel HD Graphics 620	Windows 10	2.38	56210.40	
554	HP	Notebook	8	1000 HDD	Nvidia GeForce 930MX	Windows 10	2.63	68198.40	

In [335…

```
df.select_dtypes(include=['int','float']).corr()['Price']
```

Out[335…
```
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 [336…
```
df.drop(columns=['Hybrid','Flash Storage'],inplace=True)
```

In [337…
```
df.head()
```

Out[337…

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

In [338…
```
df['Gpu'].value_counts()
```

Out[338…
```
Gpu
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: count, Length: 110, dtype: int64
```

In [339…
```
df['Gpu_brand']=df['Gpu'].apply(lambda x:x.split()[0])
```

In [340…
```
df.head()
```

Out[340…

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

In [341…
```
df['Gpu_brand'].value_counts()
```

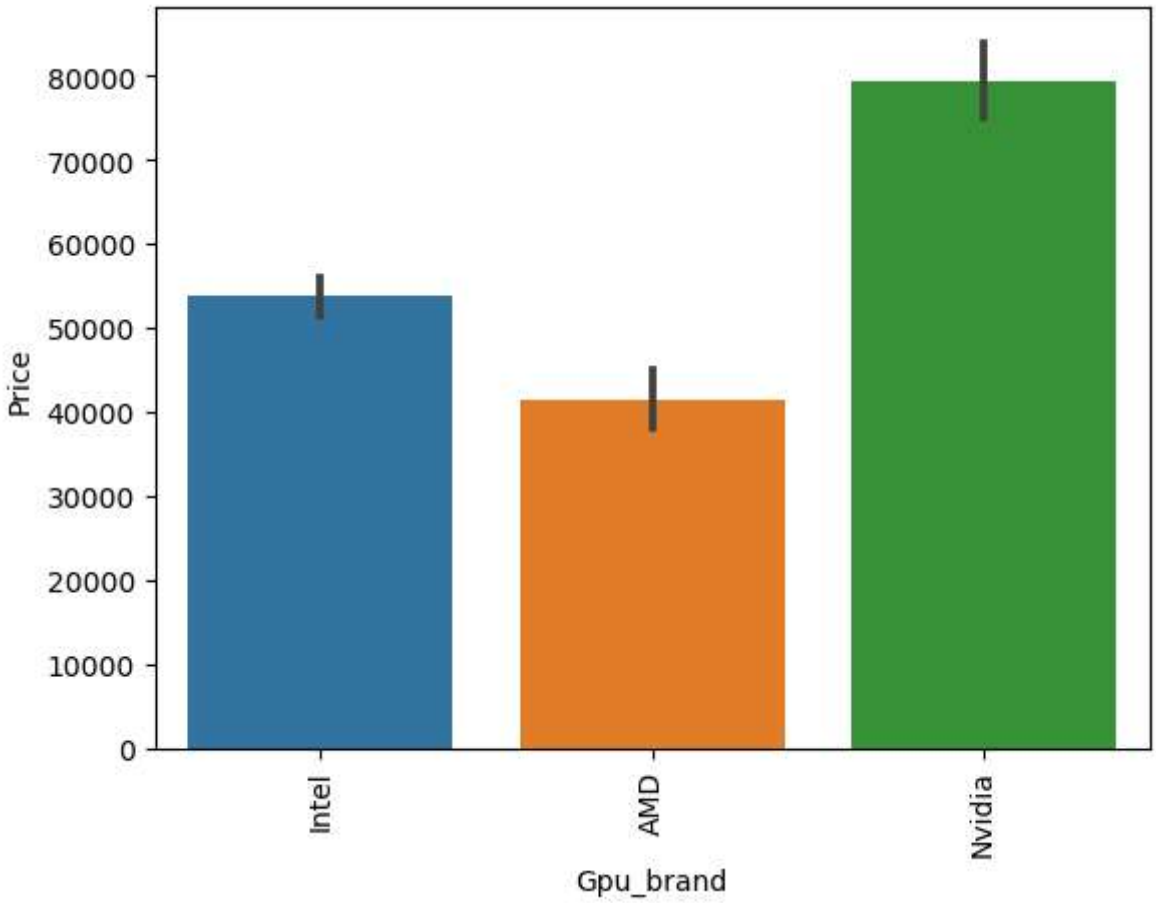
Out[341…
```
Gpu_brand
Intel     722
Nvidia    400
AMD       180
ARM         1
Name: count, dtype: int64
```

In [342…
```
df=df[df['Gpu_brand'] != 'ARM']
```

In [343…
```
df['Gpu_brand'].value_counts()
```

Out[343…
```
Gpu_brand
Intel     722
Nvidia    400
AMD       180
Name: count, dtype: int64
```

In [344…
```
sns.barplot(x=df['Gpu_brand'], y=df['Price'])
plt.xticks(rotation=90)
plt.show()
```



In [345…
```
df.drop(columns=['Gpu'],inplace=True)
```

In [346…
```
df.head()
```

Out[346...

	Company	TypeName	Ram	Memory	OpSys	Weight	Price	Touchscreen	IPS
0	Apple	Ultrabook	8	128 SSD	macOS	1.37	71378.6832	0	1
1	Apple	Ultrabook	8	128 Flash Storage	macOS	1.34	47895.5232	0	0
2	HP	Notebook	8	256 SSD	No OS	1.86	30636.0000	0	0
3	Apple	Ultrabook	16	512 SSD	macOS	1.83	135195.3360	0	1
4	Apple	Ultrabook	8	256 SSD	macOS	1.37	96095.8080	0	1

In [347...

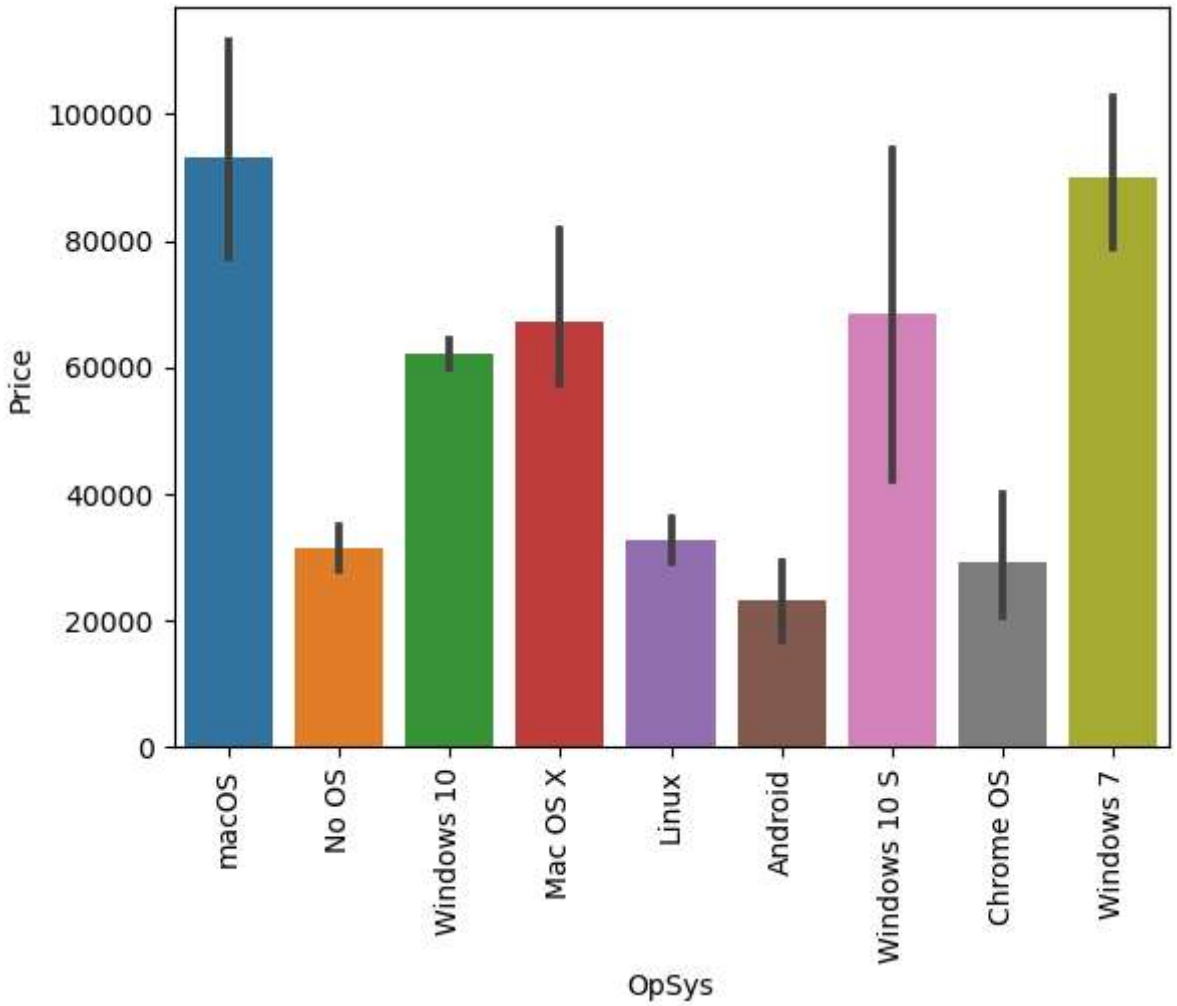
```
df['OpSys'].value_counts()
```

Out[347...

```
OpSys
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: count, dtype: int64
```

In [348...

```
sns.barplot(x=df['OpSys'],y=df['Price'])
plt.xticks(rotation=90)
plt.show()
```



In [349…
```
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 'Other/No OS/Linux'
```

In [350…
```
df['os']= df['OpSys'].apply(cat_os)
```

In [351…
```
df.head()
```

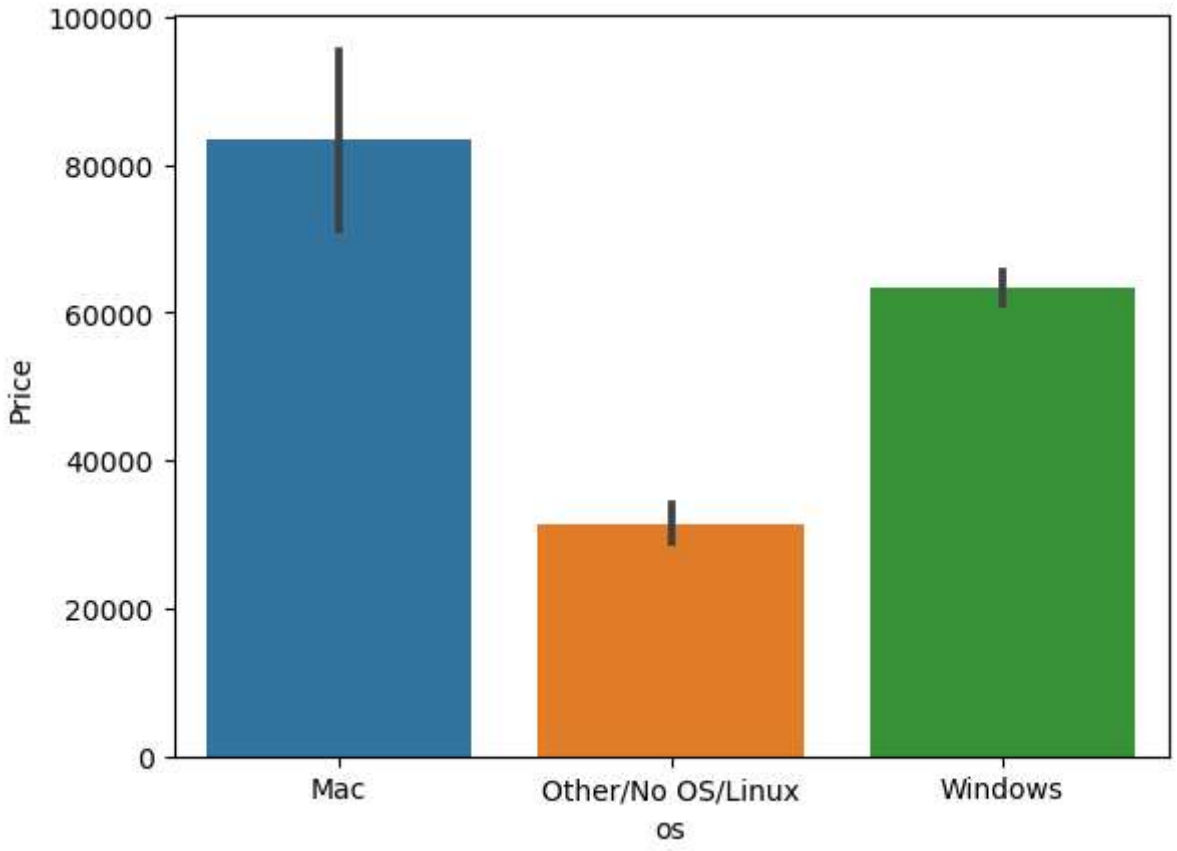
Out[351...

	Company	TypeName	Ram	Memory	OpSys	Weight	Price	Touchscreen	IPS
0	Apple	Ultrabook	8	128 SSD	macOS	1.37	71378.6832	0	1
1	Apple	Ultrabook	8	128 Flash Storage	macOS	1.34	47895.5232	0	0
2	HP	Notebook	8	256 SSD	No OS	1.86	30636.0000	0	0
3	Apple	Ultrabook	16	512 SSD	macOS	1.83	135195.3360	0	1
4	Apple	Ultrabook	8	256 SSD	macOS	1.37	96095.8080	0	1

In [352...
```
df.drop(columns=['OpSys'], inplace=True)
```

In [353...
```
sns.barplot(x=df['os'],y=df['Price'])
```

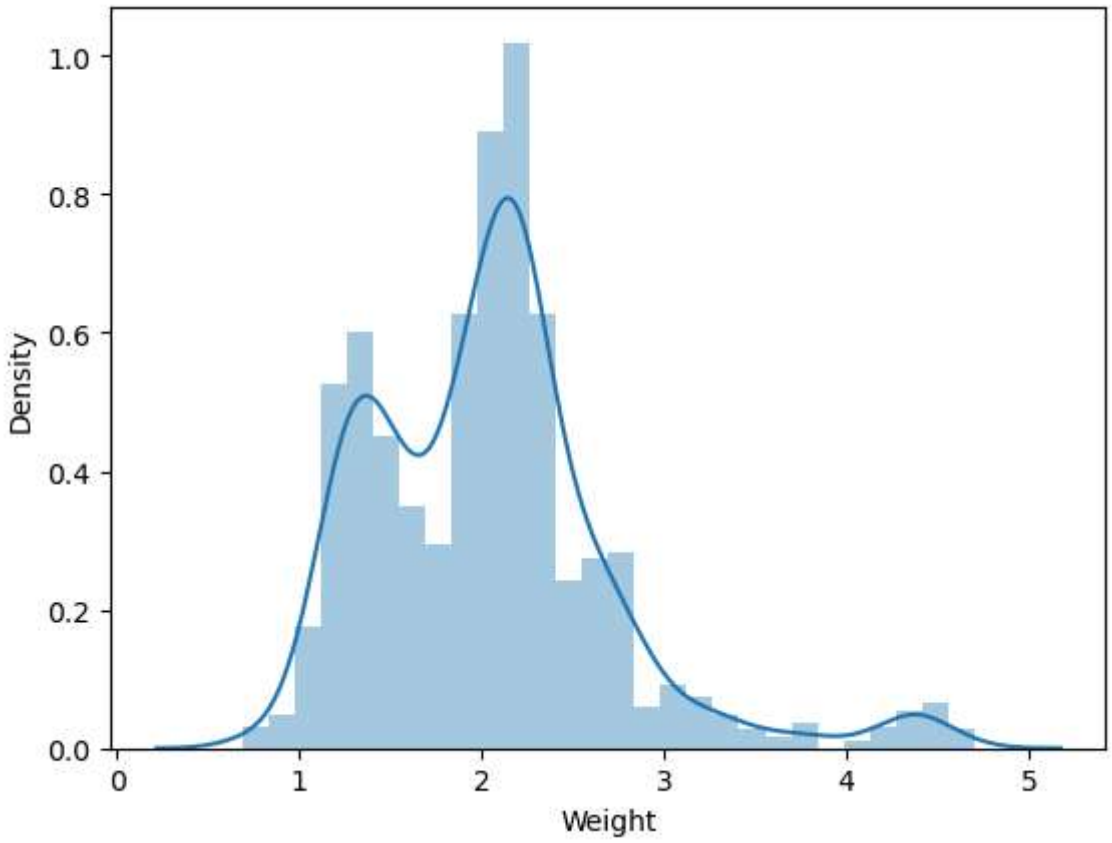
Out[353... <Axes: xlabel='os', ylabel='Price'>



In [354…
```
sns.distplot(df['Weight'])
```

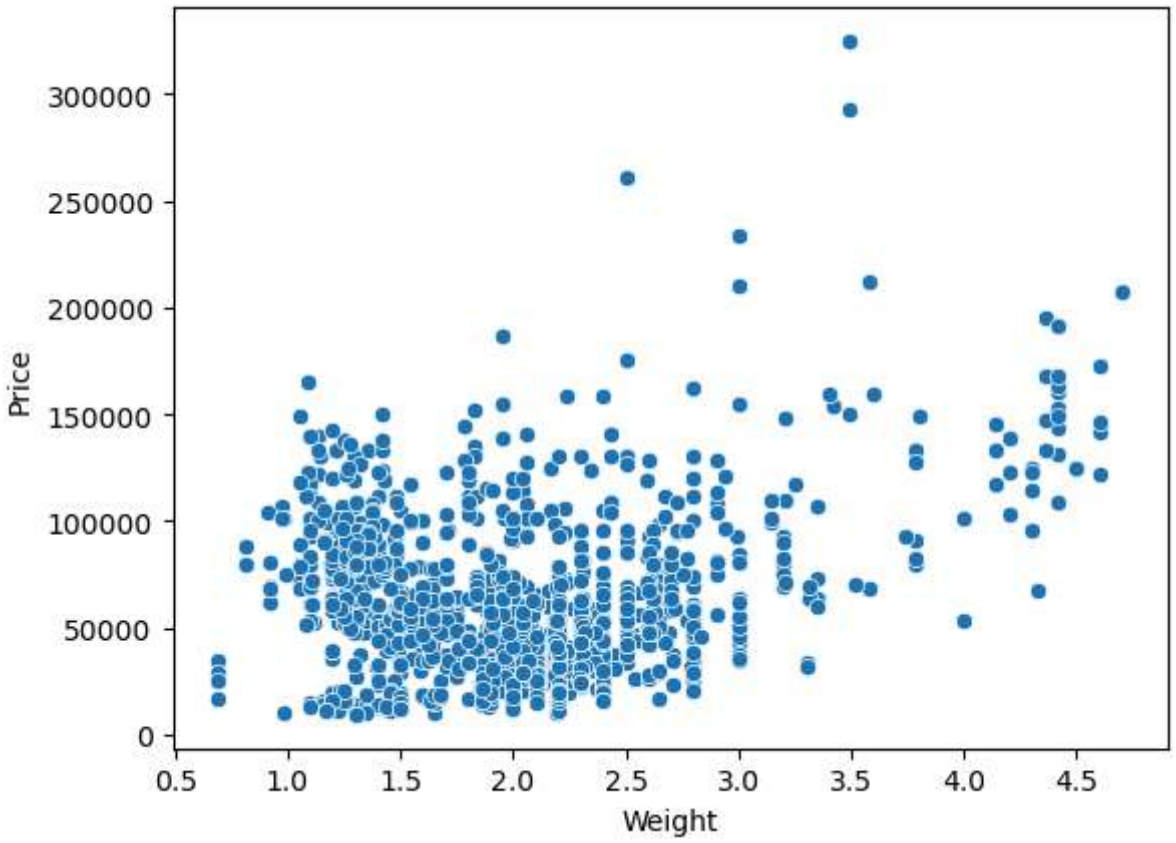
```
C:\Users\adity\AppData\Local\Temp\ipykernel_10220\1125578356.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Weight'])
```

Out[354… <Axes: xlabel='Weight', ylabel='Density'>



In [355…
```
sns.scatterplot(x=df['Weight'], y=df['Price'])
```

Out[355… <Axes: xlabel='Weight', ylabel='Price'>



In [356…
```
from sklearn.preprocessing import LabelEncoder

def LE(col):
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
```

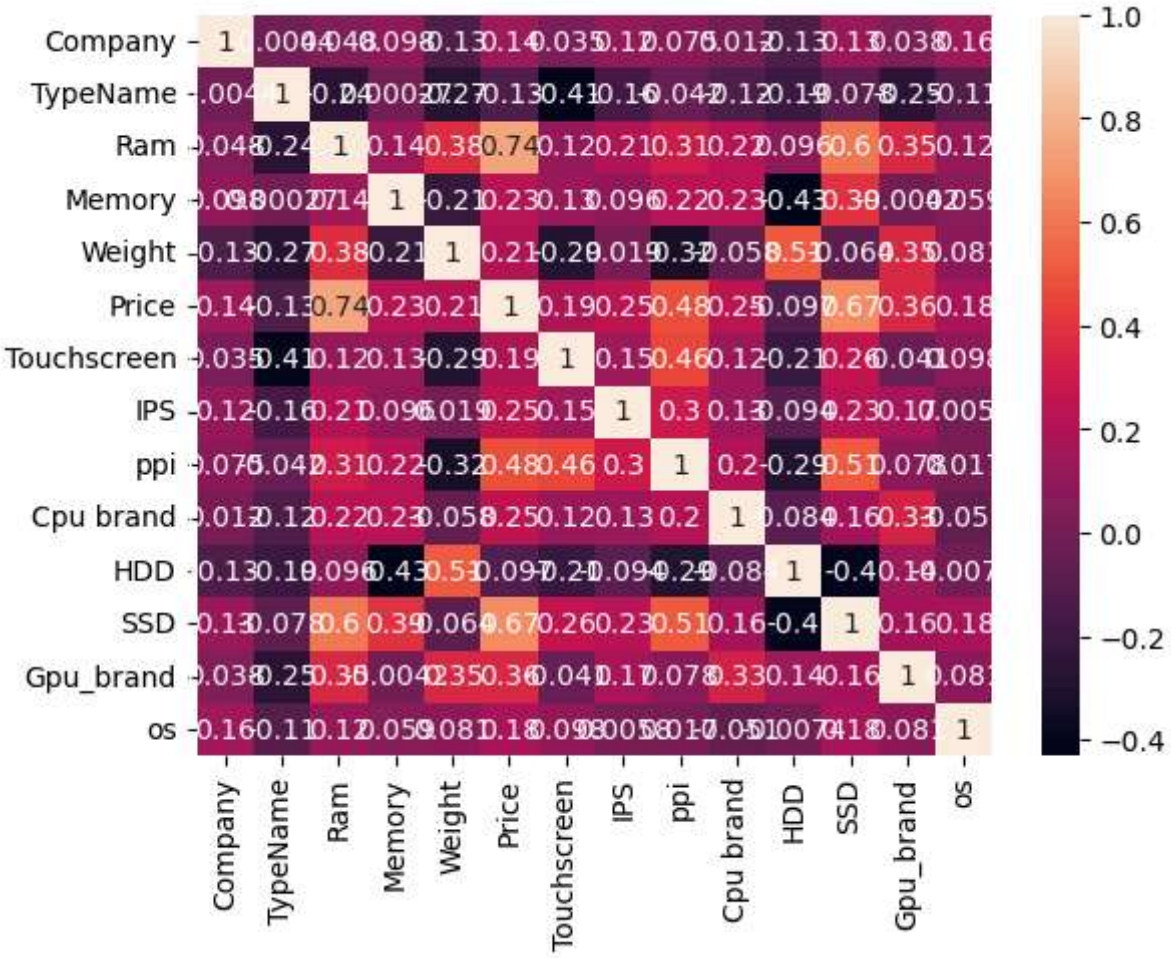
In [357…
```
for col in df.select_dtypes(include=['object']):
    LE(col)
```

In [358…
```
df.corr()['Price']
```

Out[358…
```
Company         0.141705
TypeName       -0.128608
Ram             0.742905
Memory          0.233902
Weight          0.209867
Price           1.000000
Touchscreen     0.192917
IPS             0.253320
ppi             0.475368
Cpu brand       0.245597
HDD            -0.096891
SSD             0.670660
Gpu_brand       0.355788
os              0.176937
Name: Price, dtype: float64
```

In [359…
```
sns.heatmap(df.corr(), annot=True)
```

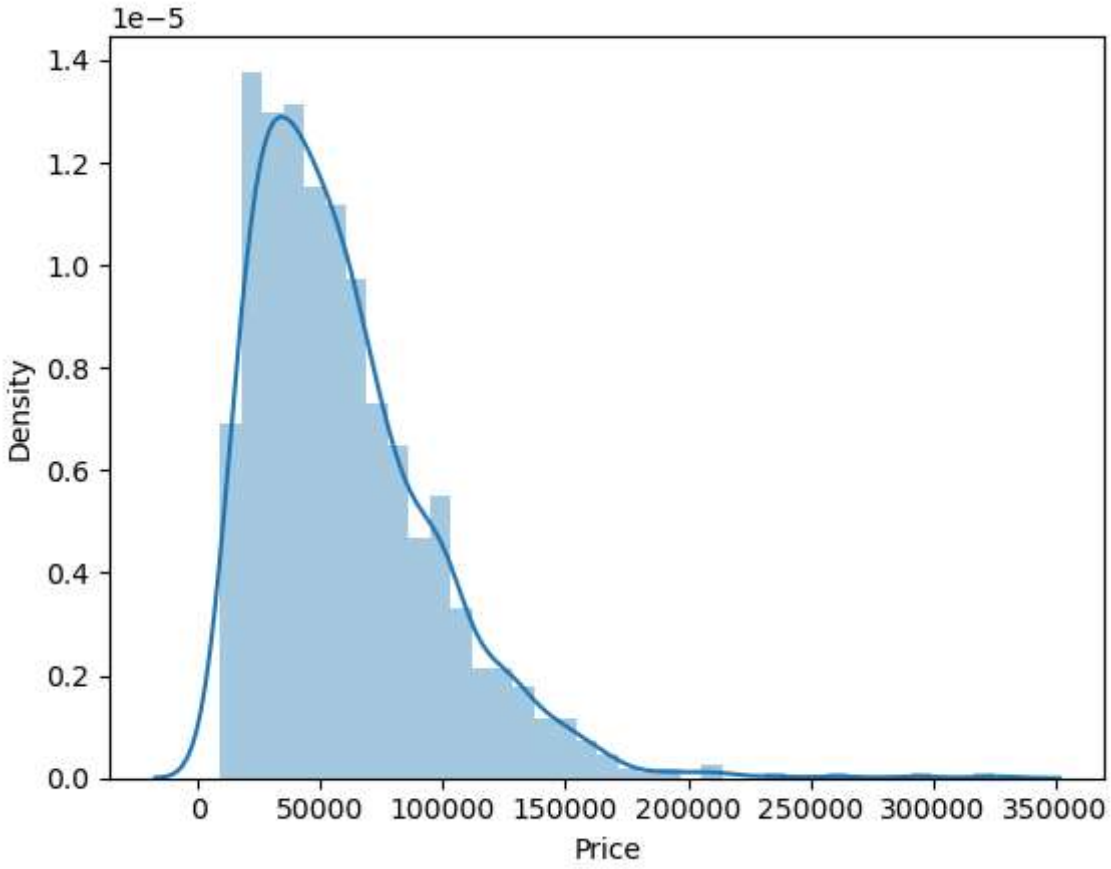
Out[359… <Axes: >



In [360…
```
sns.distplot(df['Price'])
```

```
C:\Users\adity\AppData\Local\Temp\ipykernel_10220\834922981.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Price'])
```

Out[360… <Axes: xlabel='Price', ylabel='Density'>



In [361…
```
x=df.drop(columns=['Price'])
y=df['Price']
```

In [272…
```
x
```

Out[272...

	Company	TypeName	Ram	Memory	Weight	Touchscreen	IPS	ppi	Cpu brand
0	1	4	8	7	1.37	0	1	226.983005	2
1	1	4	8	5	1.34	0	0	127.677940	2
2	7	3	8	16	1.86	0	0	141.211998	2
3	1	4	16	28	1.83	0	1	220.534624	3
4	1	4	8	16	1.37	0	1	226.983005	2
...	...	...	...	...	...	...	...	...	...
1298	10	0	4	7	1.80	1	1	157.350512	3
1299	10	0	16	28	1.30	1	1	276.053530	3
1300	10	3	2	34	1.50	0	0	111.935204	4
1301	7	3	6	0	2.19	0	0	100.454670	3
1302	2	3	4	25	2.20	0	0	100.454670	4

1302 rows × 13 columns

In [362...
```
y
```

Out[362...
```
0        71378.6832
1        47895.5232
2        30636.0000
3       135195.3360
4        96095.8080
            ...
1298     33992.6400
1299     79866.7200
1300     12201.1200
1301     40705.9200
1302     19660.3200
Name: Price, Length: 1302, dtype: float64
```

In [363...
```
from sklearn.preprocessing import PowerTransformer
```

In [364...
```
pt=PowerTransformer()
```

In [365...
```
pt
```

Out[365...
```
PowerTransformer()
```

In [366...
```
x=pt.fit_transform(x)
```

In [367…
```
x
```

Out[367…
```
array([[-1.45871097,  1.27608109,  0.22515877, ...,  0.29304511,
        -0.28172085, -2.52724752],
       [-1.45871097,  1.27608109,  0.22515877, ..., -1.30360366,
        -0.28172085, -2.52724752],
       [ 0.24706545,  0.25214789,  0.22515877, ...,  0.70828308,
        -0.28172085, -2.52026853],
       ...,
       [ 0.88904631,  0.25214789, -2.62657482, ..., -1.30360366,
        -0.28172085,  0.39665252],
       [ 0.24706545,  0.25214789, -0.34298048, ..., -1.30360366,
        -1.76960149,  0.39665252],
       [-1.09916967,  0.25214789, -1.17761125, ..., -1.30360366,
        -0.28172085,  0.39665252]])
```

In [370…
```
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.tree import DecisionTreeRegressor, DecisionTreeClassifier
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
```

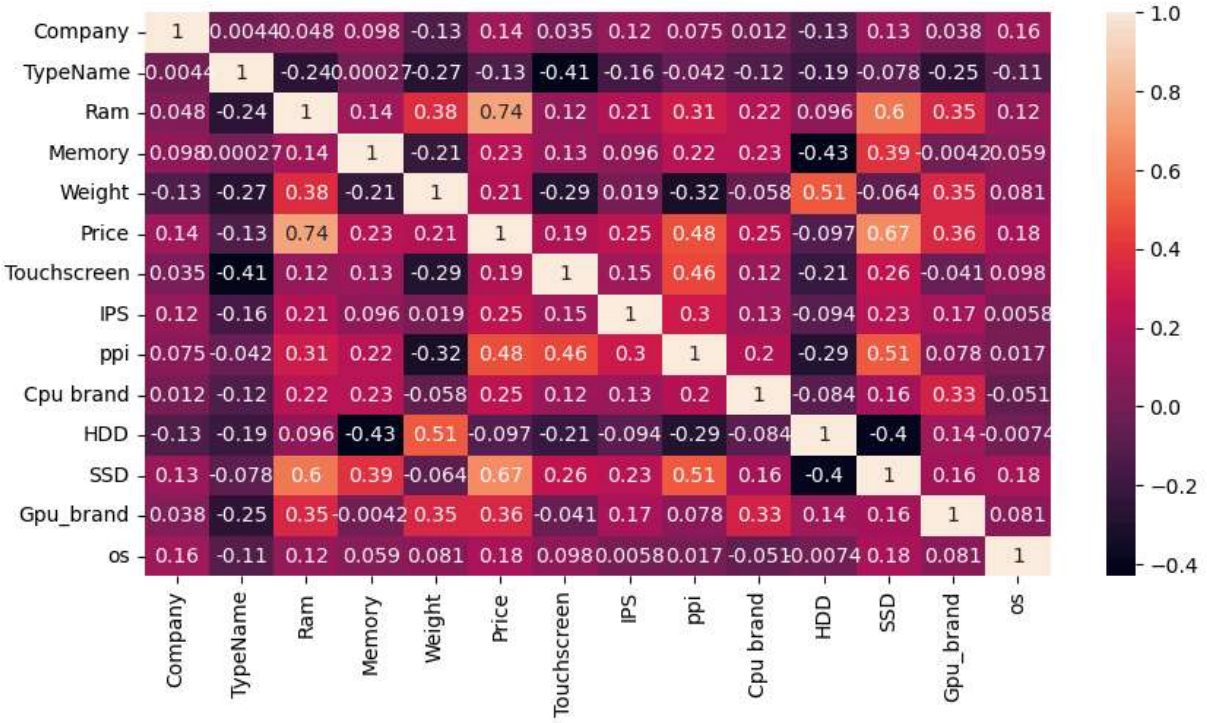
In [498…
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=45)
```

In [499…
```
x_train
```

Out[499…
```
array([[ 1.99595152,  0.25214789, -1.17761125, ..., -1.30360366,
        -0.28172085,  0.39665252],
       [-1.8999521 ,  0.25214789,  0.22515877, ...,  0.70828308,
         1.3048268 ,  0.39665252],
       [-1.8999521 ,  0.25214789, -1.17761125, ..., -1.30360366,
        -0.28172085, -2.52026853],
       ...,
       [ 0.88904631,  0.25214789, -0.34298048, ..., -1.30360366,
        -0.28172085,  0.39665252],
       [-1.09916967, -1.82071983,  0.22515877, ...,  0.70828308,
        -0.28172085,  0.39665252],
       [-0.49859984,  1.27608109,  0.22515877, ...,  0.70828308,
        -0.28172085,  0.39665252]])
```

In [500…
```
plt.figure(figsize=(10,5))
sns.heatmap(data=df.corr(),annot=True)
```

Out[500…
```
<Axes: >
```



LinearRegression()

In [501…
```
LR=LinearRegression()
```

In [502…
```
LR
```

Out[502…
```
LinearRegression
LinearRegression()
```

In [503…
```
LR.fit(x_train,y_train)
```

Out[503…
```
LinearRegression
LinearRegression()
```

In [504…
```
LR.score(x_train,y_train) , LR.score(x_test,y_test)
```

Out[504…
```
(0.6612725630600408, 0.6308615868313838)
```

In [505…
```
LR_pred=LR.predict(x_test)
```

Lasso

In [506…
```
L1=Lasso()
```

In [507…
```
L1
```

Out[507…
```
Lasso()
```

In [508…
```
L1.fit(x_train,y_train)
```

Out[508…
```
Lasso()
```

In [509…
```
L1.score(x_train,y_train), L1.score(x_test,y_test)
```

Out[509…
```
(0.6612725493283969, 0.6308676581981095)
```

In [510…
```
L1.pred=L1.predict(x_test)
```

Ridge

In [511…
```
L2=Ridge()
```

In [512…
```
L2
```

Out[512…
```
Ridge()
```

In [513…
```
L2.fit(x_train,y_train)
```

Out[513…
```
Ridge()
```

In [514…
```
L2.score(x_train,y_train), L2.score(x_test,y_test)
```

Out[514…
```
(0.6612721665895192, 0.6309665561886026)
```

Decision Tree

In [515…
```
dt=DecisionTreeRegressor(max_depth=5)
```

In [516…
```
dt
```

Out[516…
```
DecisionTreeRegressor(max_depth=5)
```

In [517…
```
dt.fit(x_train,y_train)
```

Out[517…

DecisionTreeRegressor

DecisionTreeRegressor(max_depth=5)

In [518…
```
dt.score(x_train,y_train)*100 , dt.score(x_test,y_test)*100
```

Out[518… (81.26184164192833, 58.574916672198306)

Random Forest Regressor

In [519…
```
rf=RandomForestRegressor(n_estimators=500)
rf.fit(x_train,y_train)
rf.score(x_train,y_train)*100 , rf.score(x_test,y_test)*100
```

Out[519… (97.15285526230254, 75.2693719886077)

In [ ]:

In [ ]: