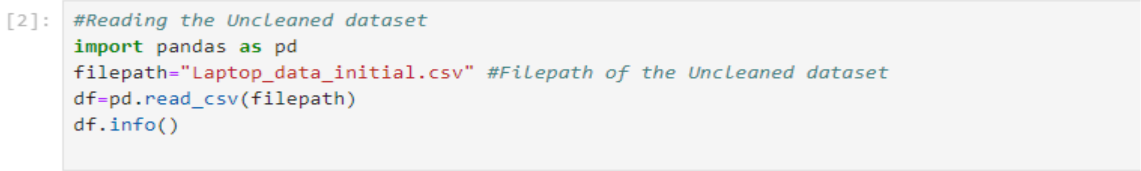
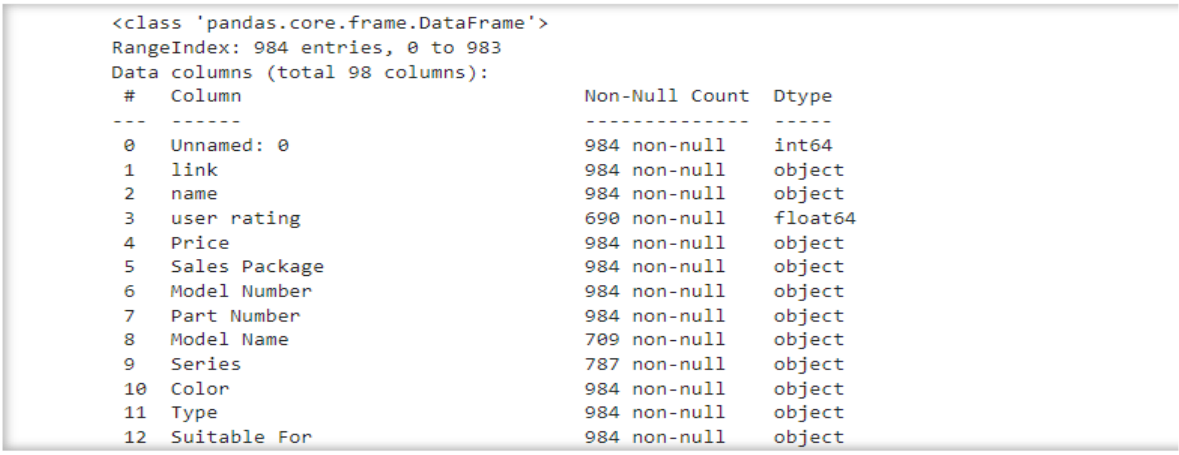
Exploratory Data Analysis of the Laptop Dataset

Author: Durvank Gade

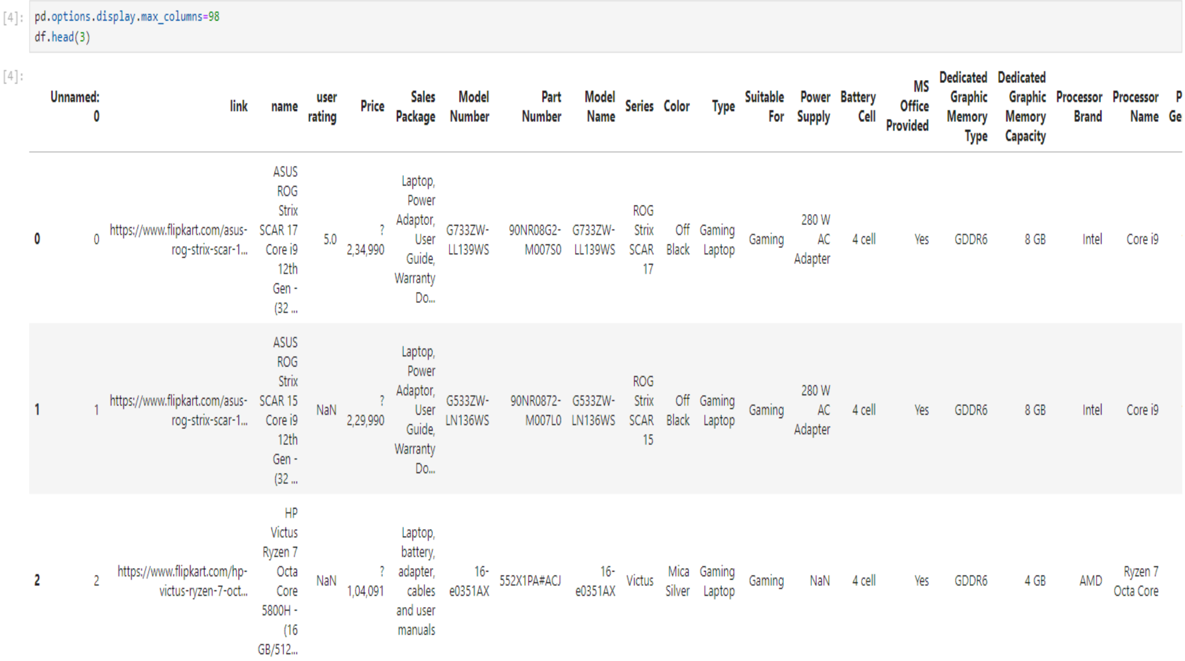
The Laptop dataset is an uncleaned dataset available at Laptop price prediction and EDA | Kaggle . This dataset contains names, user ratings, prices (In Indian Rupees) and specifications of laptops available on Flipkart. This dataset is a CSV (Comma Separated Values) file. The **read_csv()** function from pandas library was used to read the dataset.



```
[2]: #Reading the Uncleaned dataset
     import pandas as pd
     filepath="Laptop_data_initial.csv" #Filepath of the Uncleaned dataset
     df=pd.read_csv(filepath)
     df.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 984 entries, 0 to 983
Data columns (total 98 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Unnamed: 0                            984 non-null    int64
 1   link                                  984 non-null    object
 2   name                                  984 non-null    object
 3   user rating                           690 non-null    float64
 4   Price                                 984 non-null    object
 5   Sales Package                         984 non-null    object
 6   Model Number                          984 non-null    object
 7   Part Number                           984 non-null    object
 8   Model Name                            709 non-null    object
 9   Series                                787 non-null    object
 10  Color                                 984 non-null    object
 11  Type                                  984 non-null    object
 12  Suitable For                          984 non-null    object
```

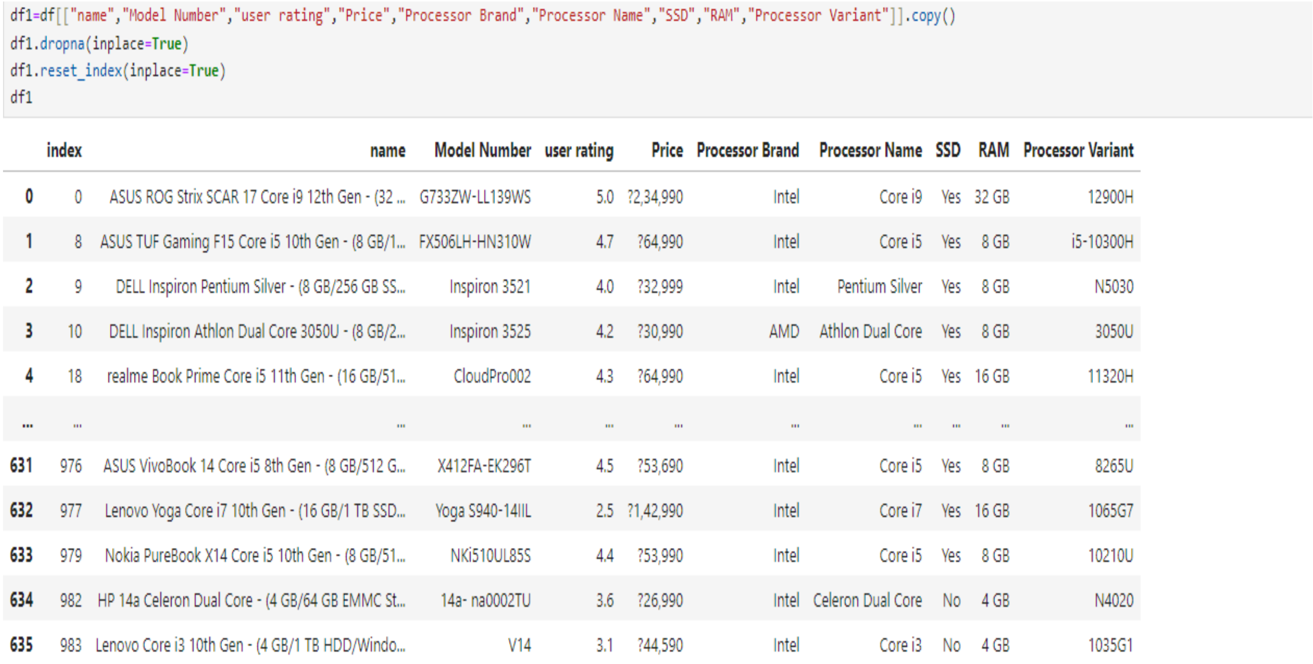


```
[4]: pd.options.display.max_columns=98
     df.head(3)
```

[4]:

	Unnamed: 0	link	name	user rating	Price	Sales Package	Model Number	Part Number	Model Name	Series	Color	Type	Suitable For	Power Supply	Battery Cell	MS Office Provided	Dedicated Graphic Memory Type	Dedicated Graphic Memory Capacity	Processor Brand	Processor Name	P Ge
0	0	https://www.flipkart.com/asus-rog-strix-scar-1...	ASUS ROG Strix SCAR 17 Core i9 12th Gen - (32 ...	5.0	? 2,34,990	Laptop, Power Adaptor, User Guide, Warranty Do...	G733ZW-LL139WS	90NR08G2-M007S0	G733ZW-LL139WS	ROG Strix SCAR 17	Off Black	Gaming Laptop	Gaming	280 W AC Adapter	4 cell	Yes	GDDR6	8 GB	Intel	Core i9	
1	1	https://www.flipkart.com/asus-rog-strix-scar-1...	ASUS ROG Strix SCAR 15 Core i9 12th Gen - (32 ...	NaN	? 2,29,990	Laptop, Power Adaptor, User Guide, Warranty Do...	G533ZW-LN136WS	90NR0872-M007L0	G533ZW-LN136WS	ROG Strix SCAR 15	Off Black	Gaming Laptop	Gaming	280 W AC Adapter	4 cell	Yes	GDDR6	8 GB	Intel	Core i9	
2	2	https://www.flipkart.com/hp-victus-ryzen-7-oct...	HP Victus Ryzen 7 Octa Core 5800H - (16 GB/512...	NaN	? 1,04,091	Laptop, battery, adapter, cables and user manuals	16-e0351AX	552X1PA#ACJ	16-e0351AX	Victus	Mica Silver	Gaming Laptop	Gaming	NaN	4 cell	Yes	GDDR6	4 GB	AMD	Ryzen 7 Octa Core	

There are 984 rows and 98 columns in the original dataset. To remove unwanted rows, a new DataFrame that contains only required columns from the original datasets can be created. The rows containing null values were removed using using **dropna()** function.

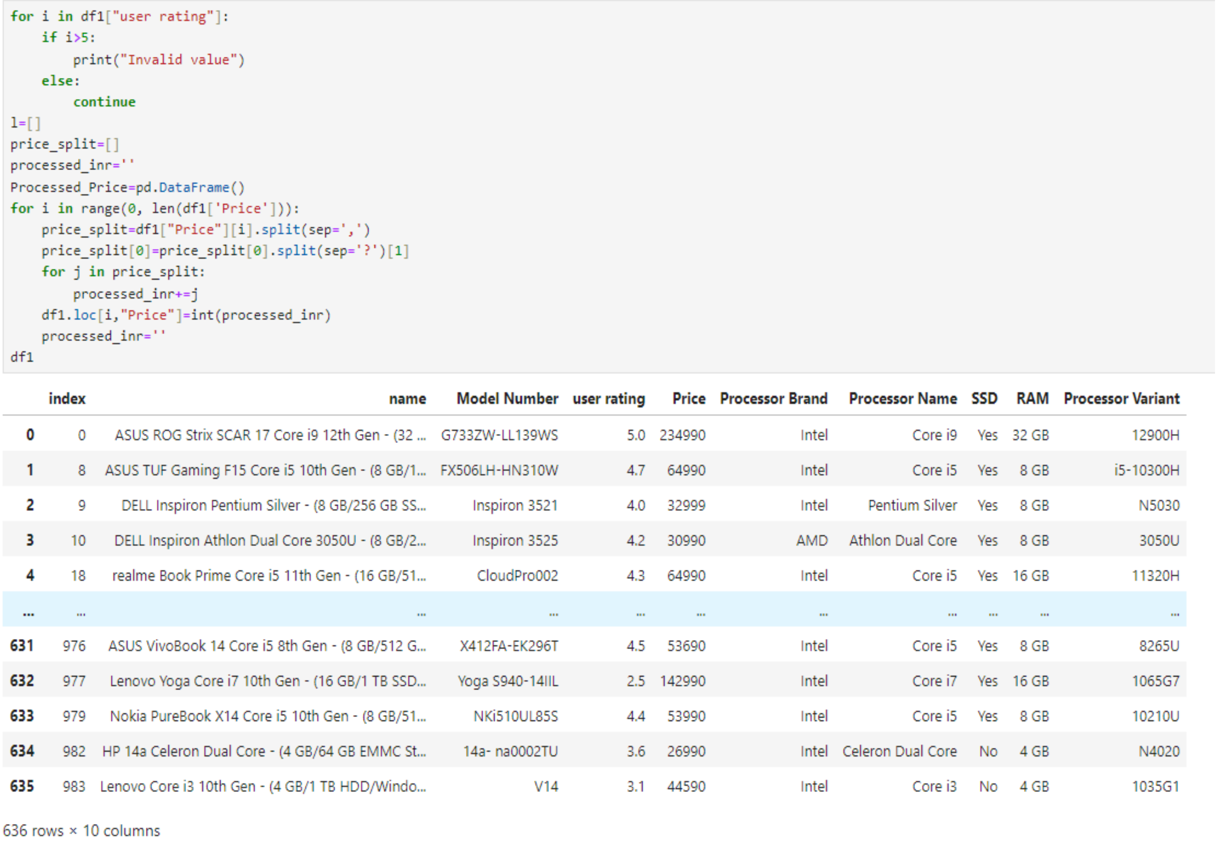


```
df1=df[["name","Model Number","user rating","Price","Processor Brand","Processor Name","SSD","RAM","Processor Variant"]].copy()
df1.dropna(inplace=True)
df1.reset_index(inplace=True)
df1
```

	index	name	Model Number	user rating	Price	Processor Brand	Processor Name	SSD	RAM	Processor Variant
0	0	ASUS ROG Strix SCAR 17 Core i9 12th Gen - (32 ...	G733ZW-LL139WS	5.0	?2,34,990	Intel	Core i9	Yes	32 GB	12900H
1	8	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/1...	FX506LH-HN310W	4.7	?64,990	Intel	Core i5	Yes	8 GB	i5-10300H
2	9	DELL Inspiron Pentium Silver - (8 GB/256 GB SS...	Inspiron 3521	4.0	?32,999	Intel	Pentium Silver	Yes	8 GB	N5030
3	10	DELL Inspiron Athlon Dual Core 3050U - (8 GB/2...	Inspiron 3525	4.2	?30,990	AMD	Athlon Dual Core	Yes	8 GB	3050U
4	18	realme Book Prime Core i5 11th Gen - (16 GB/51...	CloudPro002	4.3	?64,990	Intel	Core i5	Yes	16 GB	11320H
...	...	...	...	...	...	...	...	...	...	...
631	976	ASUS VivoBook 14 Core i5 8th Gen - (8 GB/512 G...	X412FA-EK296T	4.5	?53,690	Intel	Core i5	Yes	8 GB	8265U
632	977	Lenovo Yoga Core i7 10th Gen - (16 GB/1 TB SSD...	Yoga S940-14IIL	2.5	?1,42,990	Intel	Core i7	Yes	16 GB	1065G7
633	979	Nokia PureBook X14 Core i5 10th Gen - (8 GB/51...	NKi510UL85S	4.4	?53,990	Intel	Core i5	Yes	8 GB	10210U
634	982	HP 14a Celeron Dual Core - (4 GB/64 GB EMMC St...	14a- na0002TU	3.6	?26,990	Intel	Celeron Dual Core	No	4 GB	N4020
635	983	Lenovo Core i3 10th Gen - (4 GB/1 TB HDD/Windo...	V14	3.1	?44,590	Intel	Core i3	No	4 GB	1035G1

The “user rating” column contains the ratings given by the customers for each laptop, ranging from 0 to 5. To ensure that there are no invalid values in this column, we can use a for loop to iterate over the ratings and check if any of them are greater than 5. After running the code, we find that there are no such values, so we do not need to perform any further cleaning on this column.

The “price” column shows the price of each laptop in Indian Rupees (INR). However, some of the values have ‘?’ and ‘,’ symbols, which make them difficult to process as numerical data. To remove these symbols and convert the value into integers, we can use the following code:



```
for i in df1["user rating"]:
    if i>5:
        print("Invalid value")
    else:
        continue
l=[]
price_split=[]
processed_inr=''
Processed_Price=pd.DataFrame()
for i in range(0, len(df1['Price'])):
    price_split=df1["Price"][i].split(sep=',')
    price_split[0]=price_split[0].split(sep='?')[1]
    for j in price_split:
        processed_inr+=j
    df1.loc[i,"Price"]=int(processed_inr)
    processed_inr=''
df1
```

	index	name	Model Number	user rating	Price	Processor Brand	Processor Name	SSD	RAM	Processor Variant
0	0	ASUS ROG Strix SCAR 17 Core i9 12th Gen - (32 ...	G733ZW-LL139WS	5.0	234990	Intel	Core i9	Yes	32 GB	12900H
1	8	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/1...	FX506LH-HN310W	4.7	64990	Intel	Core i5	Yes	8 GB	i5-10300H
2	9	DELL Inspiron Pentium Silver - (8 GB/256 GB SS...	Inspiron 3521	4.0	32999	Intel	Pentium Silver	Yes	8 GB	N5030
3	10	DELL Inspiron Athlon Dual Core 3050U - (8 GB/2...	Inspiron 3525	4.2	30990	AMD	Athlon Dual Core	Yes	8 GB	3050U
4	18	realme Book Prime Core i5 11th Gen - (16 GB/51...	CloudPro002	4.3	64990	Intel	Core i5	Yes	16 GB	11320H
...	...	...	...	...	...	...	...	...	...	...
631	976	ASUS VivoBook 14 Core i5 8th Gen - (8 GB/512 G...	X412FA-EK296T	4.5	53690	Intel	Core i5	Yes	8 GB	8265U
632	977	Lenovo Yoga Core i7 10th Gen - (16 GB/1 TB SSD...	Yoga S940-14IIL	2.5	142990	Intel	Core i7	Yes	16 GB	1065G7
633	979	Nokia PureBook X14 Core i5 10th Gen - (8 GB/51...	NKi510UL85S	4.4	53990	Intel	Core i5	Yes	8 GB	10210U
634	982	HP 14a Celeron Dual Core - (4 GB/64 GB EMMC St...	14a- na0002TU	3.6	26990	Intel	Celeron Dual Core	No	4 GB	N4020
635	983	Lenovo Core i3 10th Gen - (4 GB/1 TB HDD/Windo...	V14	3.1	44590	Intel	Core i3	No	4 GB	1035G1

636 rows × 10 columns

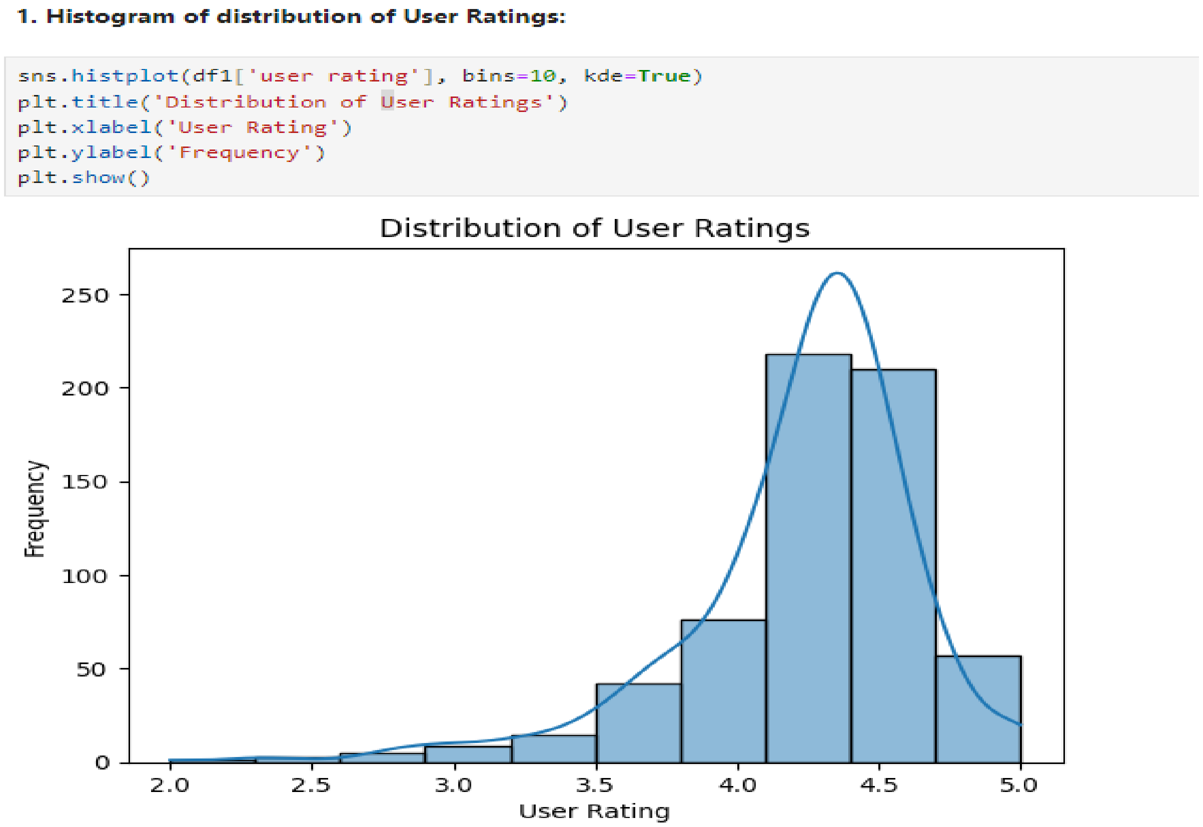
The cleaned data provides many useful insights, such as price trends, user preferences, feature correlations, and market opportunities. For example, we can:

- Examine the most common specifications of laptops and compare them with the average prices and ratings.
- Identify potential gaps or opportunities in the market based on the demand and supply of different laptop features.
- Investigate whether there is a relationship between the processor brand and the user ratings of laptops.

This can be achieved through graphical visualisation of data using Python libraries like matplotlib, seaborn, etc.

Some of the plots possible are:

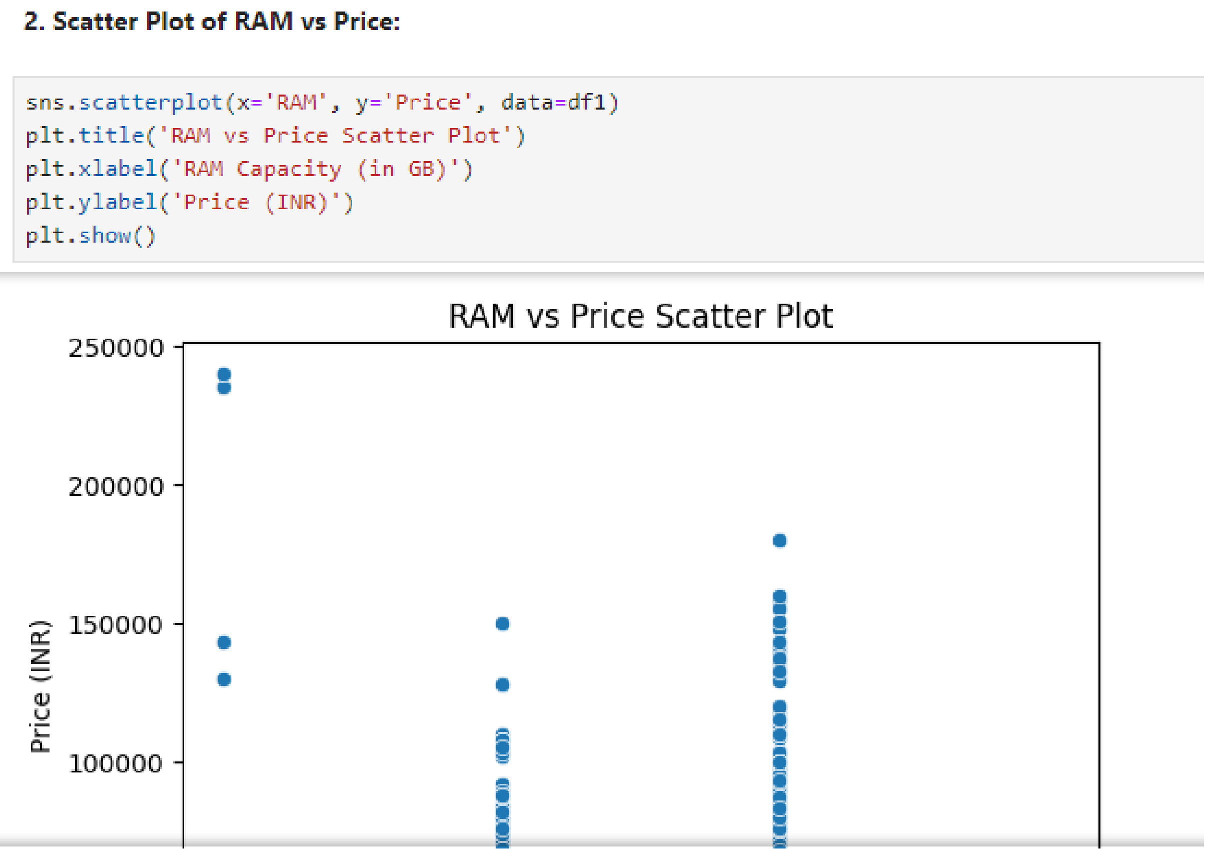
1. Histogram of User Ratings:



1. Histogram of distribution of User Ratings:

```
sns.histplot(df1['user rating'], bins=10, kde=True)
plt.title('Distribution of User Ratings')
plt.xlabel('User Rating')
plt.ylabel('Frequency')
plt.show()
```

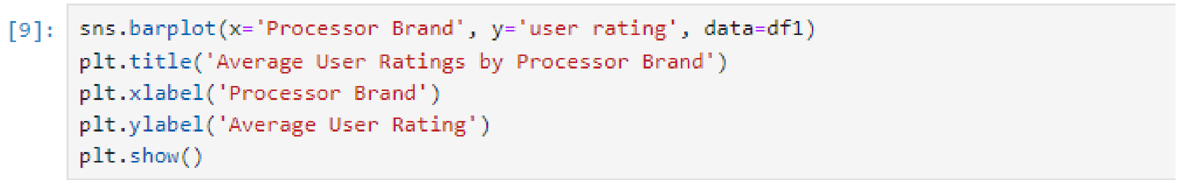
2. Scatter Plot of RAM vs Price:



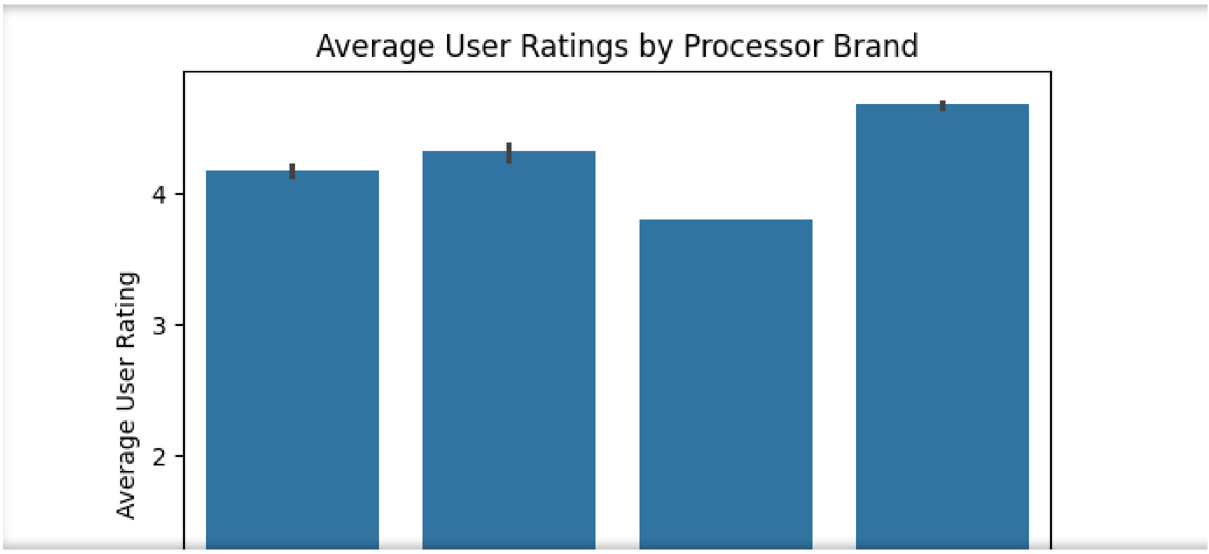
2. Scatter Plot of RAM vs Price:

```
sns.scatterplot(x='RAM', y='Price', data=df1)
plt.title('RAM vs Price Scatter Plot')
plt.xlabel('RAM Capacity (in GB)')
plt.ylabel('Price (INR)')
plt.show()
```

3.User Ratings vs Processor Brands:



```
[9]: sns.barplot(x='Processor Brand', y='user rating', data=df1)
     plt.title('Average User Ratings by Processor Brand')
     plt.xlabel('Processor Brand')
     plt.ylabel('Average User Rating')
     plt.show()
```



Average User Ratings by Processor Brand
Average User Rating