DATA FOLKZ

CAPSTONE PROJECT
DATA ANALYTICS

Web Scraping Project

Submitted By –

HEMANT PATAR

OBJECTIVE:

As per the choices given between 1. Flipkart.com 2. Ajio.com 3. Snapdeal.com I chose Snapdeal as my target website to scrape in this project.

I chose Snapdeal as it was a dynamic website with scrolling pages which I wanted to learn and practice in this project.

Apart from scraping of details of a specific product range data cleaning and formatting is done with visualizations.

SCRAPING:

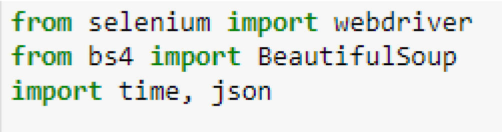
I chose mobile phones as my target product as people now a days prefer buying it online with discounted price. And I am a tech enthusiast also.

After inspecting the website script we identify the tags to grab the values (product name, product rating, product price, some general specifications like (processor, memory, RAM, camera, screen, color))

We see all the values we need are inside the link to the specific product.

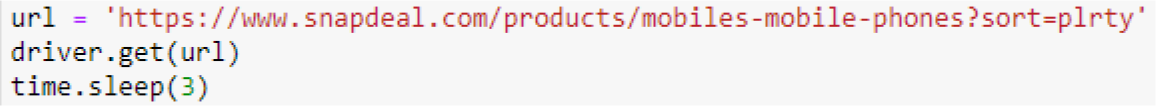
So we need to scrape all the product links first.

Let’s import the required libraries.



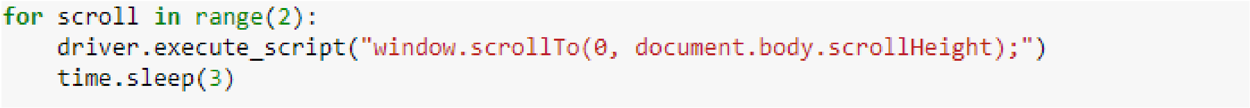
```
from selenium import webdriver
from bs4 import BeautifulSoup
import time, json
```

We need to use selenium to automate the scrolling and grab the product links.



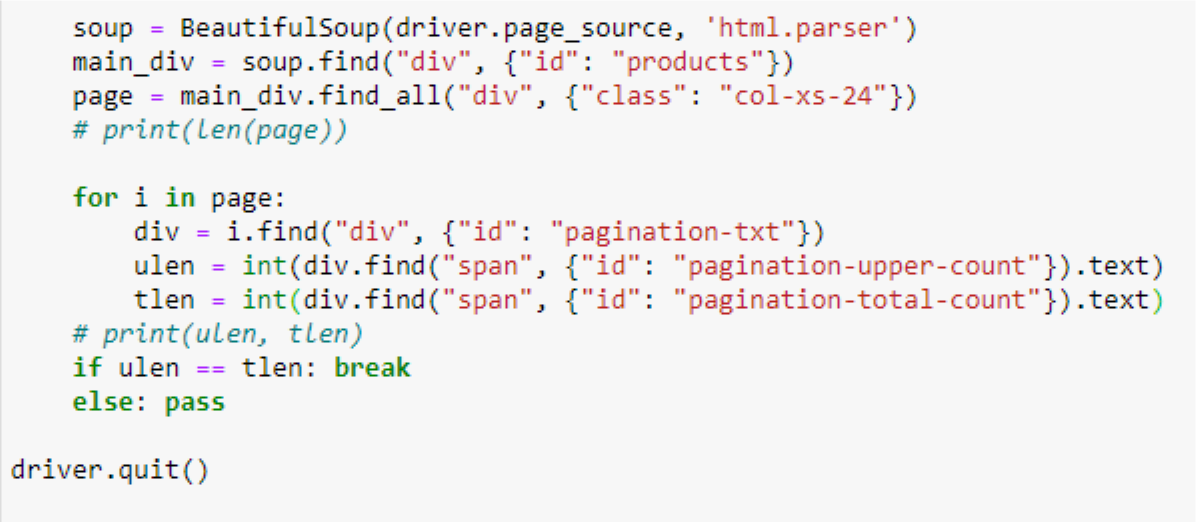
```
url = 'https://www.snapdeal.com/products/mobiles-mobile-phones?sort=plrty'
driver.get(url)
time.sleep(3)
```

Above is the url of all the mobile phones in snapdeal. We use timer to let the page load while scrolling done by automated chrome.



```
for scroll in range(2):
    driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
    time.sleep(3)
```

Above is the code to automate scrolling using selenium.



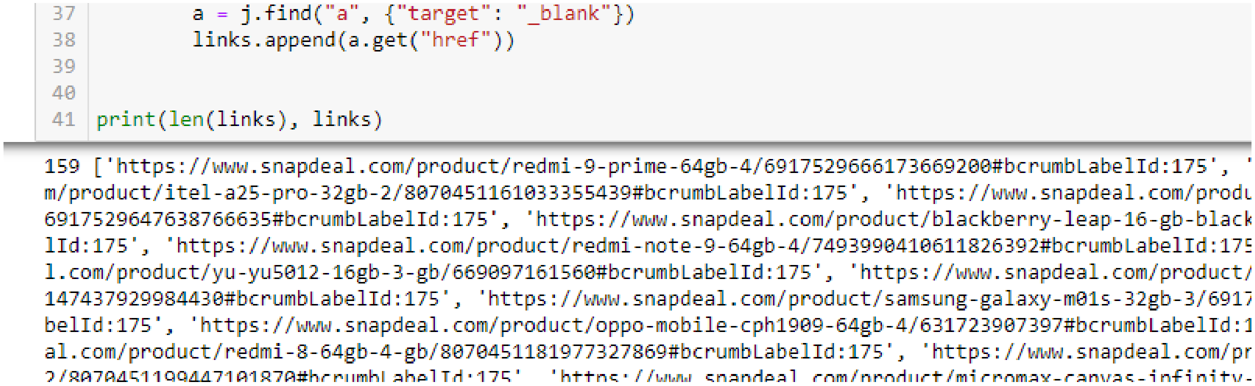
```
    soup = BeautifulSoup(driver.page_source, 'html.parser')
    main_div = soup.find("div", {"id": "products"})
    page = main_div.find_all("div", {"class": "col-xs-24"})
    # print(len(page))

    for i in page:
        div = i.find("div", {"id": "pagination-txt"})
        ulen = int(div.find("span", {"id": "pagination-upper-count"}).text)
        tlen = int(div.find("span", {"id": "pagination-total-count"}).text)
    # print(ulen, tlen)
    if ulen == tlen: break
    else: pass

driver.quit()
```

We use BeautifulSoup to grab the links via above tags. Here we make use of page markers to propagate throughout the page to end of page.

And quit the driver after reading the end of page.



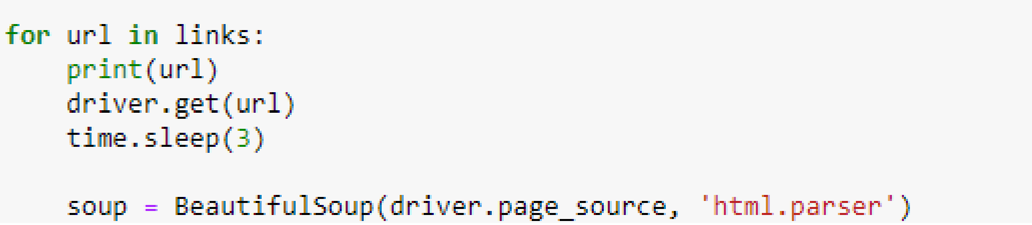
```
        a = j.find("a", {"target": "_blank"})
        links.append(a.get("href"))


print(len(links), links)
```

```
159 ['https://www.snapdeal.com/product/redmi-9-prime-64gb-4/6917529666173669200#bcrumbLabelId:175', '
m/product/itel-a25-pro-32gb-2/8070451161033355439#bcrumbLabelId:175', 'https://www.snapdeal.com/produ
6917529647638766635#bcrumbLabelId:175', 'https://www.snapdeal.com/product/blackberry-leap-16-gb-black
lId:175', 'https://www.snapdeal.com/product/redmi-note-9-64gb-4/7493990410611826392#bcrumbLabelId:175
l.com/product/yu-yu5012-16gb-3-gb/669097161560#bcrumbLabelId:175', 'https://www.snapdeal.com/product/
147437929984430#bcrumbLabelId:175', 'https://www.snapdeal.com/product/samsung-galaxy-m01s-32gb-3/6917
belId:175', 'https://www.snapdeal.com/product/oppo-mobile-cph1909-64gb-4/631723907397#bcrumbLabelId:1
al.com/product/redmi-8-64gb-4-gb/8070451181977327869#bcrumbLabelId:175', 'https://www.snapdeal.com/pr
2/8070451199447101870#bcrumbLabelId:175', 'https://www.snapdeal.com/product/micromax-canvas-infinity-
```

Now save the grabbed links as list. Here we grabbed 159 links that means 159 products(mobile phones) were identified.

Now with the help of these links we will scrape each product details one by one.

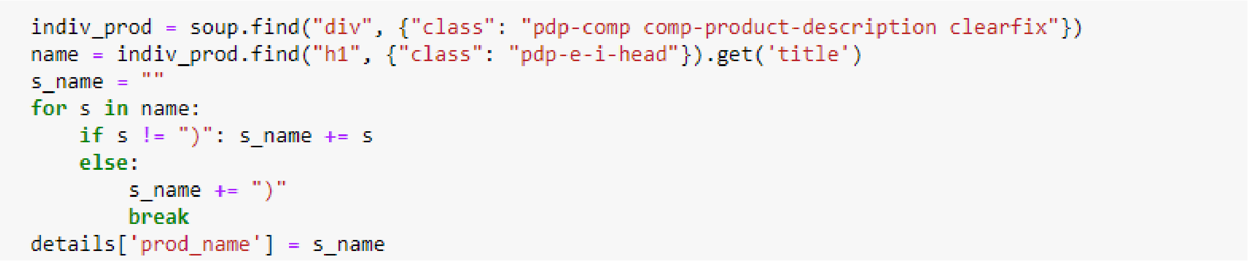


```
for url in links:
    print(url)
    driver.get(url)
    time.sleep(3)

    soup = BeautifulSoup(driver.page_source, 'html.parser')
```

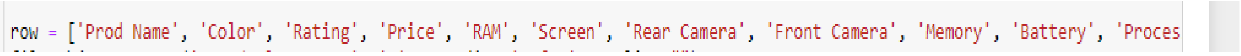
In the above code we parse all the links from the list generated above one by one .

After inspecting one link from the list of links we identify the tags we need to use to grab the product specifications.



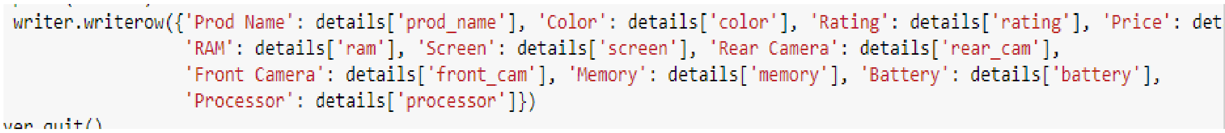
```
indiv_prod = soup.find("div", {"class": "pdp-comp comp-product-description clearfix"})
name = indiv_prod.find("h1", {"class": "pdp-e-i-head"}).get('title')
s_name = ""
for s in name:
    if s != ")": s_name += s
    else:
        s_name += ")"
        break
details['prod_name'] = s_name
```

In the above code we grab the product name from the soup and save as list . Like this all the other features are also grabbed.



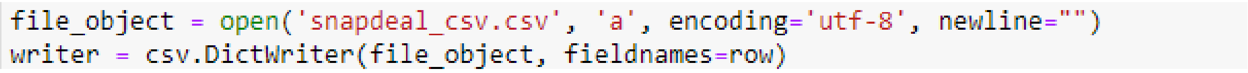
```
row = ['Prod Name', 'Color', 'Rating', 'Price', 'RAM', 'Screen', 'Rear Camera', 'Front Camera', 'Memory', 'Battery', 'Proces
```

We save all the features as dictionary.

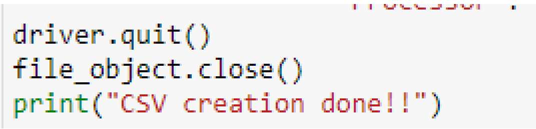


```
writer.writerow({'Prod Name': details['prod_name'], 'Color': details['color'], 'Rating': details['rating'], 'Price': det
                 'RAM': details['ram'], 'Screen': details['screen'], 'Rear Camera': details['rear_cam'],
                 'Front Camera': details['front_cam'], 'Memory': details['memory'], 'Battery': details['battery'],
                 'Processor': details['processor']})
ver.quit()
```

Now save all the features as csv with file name snapdeal_csv.csv



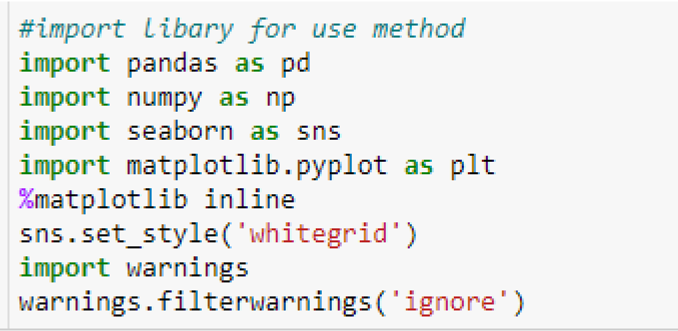
```
file_object = open('snapdeal_csv.csv', 'a', encoding='utf-8', newline="")
writer = csv.DictWriter(file_object, fieldnames=row)
```



```
driver.quit()
file_object.close()
print("CSV creation done!!")
```

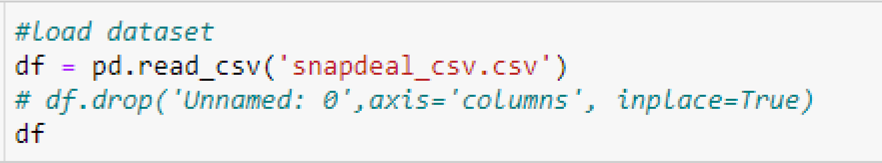
Close the csv file.

Our scrapped data of all the mobile phones in snapdeal is saved as csv for next part of project.



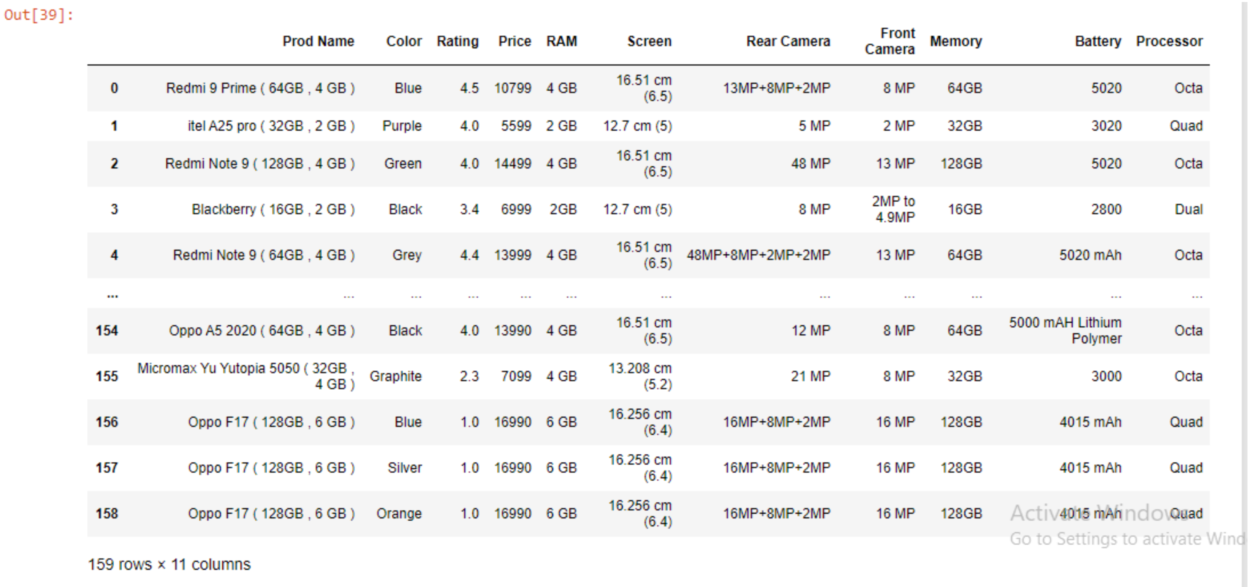
```
#import libary for use method
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set_style('whitegrid')
import warnings
warnings.filterwarnings('ignore')
```

Import libraries for data processing of scrapped data.



```
#Load dataset
df = pd.read_csv('snapdeal_csv.csv')
# df.drop('Unnamed: 0',axis='columns', inplace=True)
df
```

Load dataset as df in Jupyter for processing and EDA.



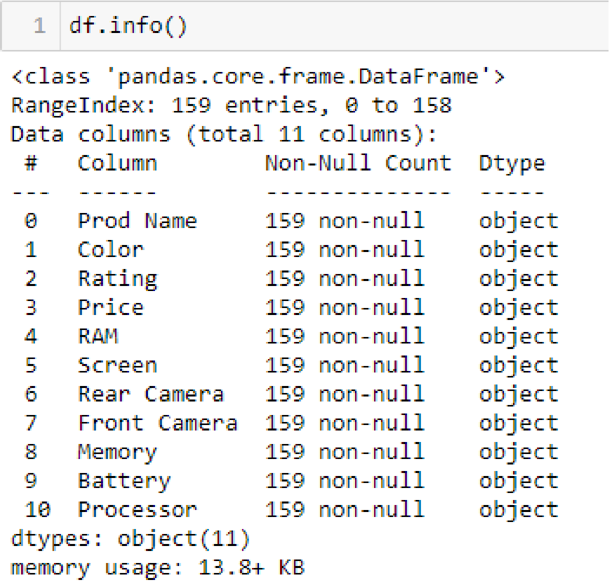
Out[39]:

	Prod Name	Color	Rating	Price	RAM	Screen	Rear Camera	Front Camera	Memory	Battery	Processor
0	Redmi 9 Prime (64GB , 4 GB)	Blue	4.5	10799	4 GB	16.51 cm (6.5)	13MP+8MP+2MP	8 MP	64GB	5020	Octa
1	itel A25 pro (32GB , 2 GB)	Purple	4.0	5599	2 GB	12.7 cm (5)	5 MP	2 MP	32GB	3020	Quad
2	Redmi Note 9 (128GB , 4 GB)	Green	4.0	14499	4 GB	16.51 cm (6.5)	48 MP	13 MP	128GB	5020	Octa
3	Blackberry (16GB , 2 GB)	Black	3.4	6999	2GB	12.7 cm (5)	8 MP	2MP to 4.9MP	16GB	2800	Dual
4	Redmi Note 9 (64GB , 4 GB)	Grey	4.4	13999	4 GB	16.51 cm (6.5)	48MP+8MP+2MP+2MP	13 MP	64GB	5020 mAh	Octa
...	...	...	...	...	...	...	...	...	...	...	...
154	Oppo A5 2020 (64GB , 4 GB)	Black	4.0	13990	4 GB	16.51 cm (6.5)	12 MP	8 MP	64GB	5000 mAH Lithium Polymer	Octa
155	Micromax Yu Yutopia 5050 (32GB , 4 GB)	Graphite	2.3	7099	4 GB	13.208 cm (5.2)	21 MP	8 MP	32GB	3000	Octa
156	Oppo F17 (128GB , 6 GB)	Blue	1.0	16990	6 GB	16.256 cm (6.4)	16MP+8MP+2MP	16 MP	128GB	4015 mAh	Quad
157	Oppo F17 (128GB , 6 GB)	Silver	1.0	16990	6 GB	16.256 cm (6.4)	16MP+8MP+2MP	16 MP	128GB	4015 mAh	Quad
158	Oppo F17 (128GB , 6 GB)	Orange	1.0	16990	6 GB	16.256 cm (6.4)	16MP+8MP+2MP	16 MP	128GB	4015 mAh	Quad

159 rows × 11 columns

Above is the glance of our scrapped data from snapdeal.

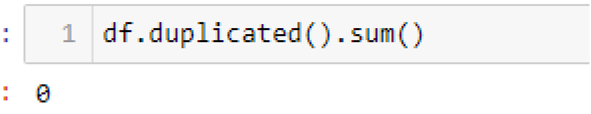
We have 159 rows i.e. 159 mobile phones and 11 columns as features of each mobile phone.



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159 entries, 0 to 158
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Prod Name     159 non-null    object
 1   Color         159 non-null    object
 2   Rating        159 non-null    object
 3   Price         159 non-null    object
 4   RAM           159 non-null    object
 5   Screen        159 non-null    object
 6   Rear Camera   159 non-null    object
 7   Front Camera  159 non-null    object
 8   Memory        159 non-null    object
 9   Battery       159 non-null    object
 10  Processor     159 non-null    object
dtypes: object(11)
memory usage: 13.8+ KB
```

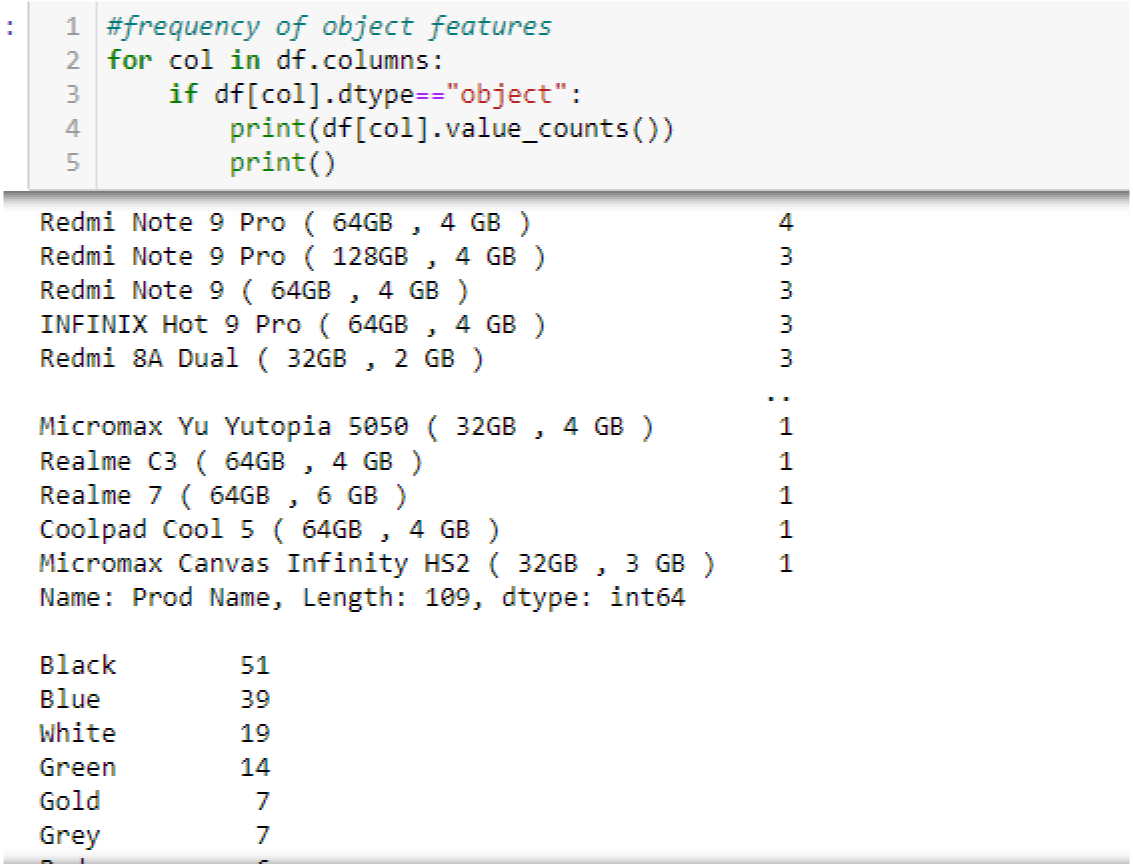
We have no null values in any of the features.



```
: df.duplicated().sum()
```

```
: 0
```

We have no duplicate or repeated products.

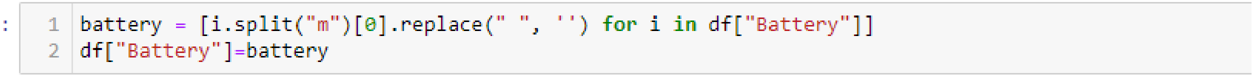


```
: #frequency of object features
  for col in df.columns:
      if df[col].dtype=="object":
          print(df[col].value_counts())
          print()
```

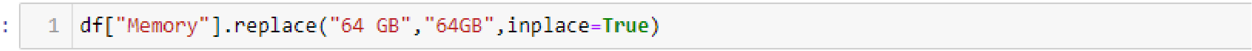
```
Redmi Note 9 Pro ( 64GB , 4 GB )                  4
Redmi Note 9 Pro ( 128GB , 4 GB )                 3
Redmi Note 9 ( 64GB , 4 GB )                      3
INFINIX Hot 9 Pro ( 64GB , 4 GB )                 3
Redmi 8A Dual ( 32GB , 2 GB )                     3
                                                 ..
Micromax Yu Yutopia 5050 ( 32GB , 4 GB )          1
Realme C3 ( 64GB , 4 GB )                         1
Realme 7 ( 64GB , 6 GB )                          1
Coolpad Cool 5 ( 64GB , 4 GB )                    1
Micromax Canvas Infinity HS2 ( 32GB , 3 GB )      1
Name: Prod Name, Length: 109, dtype: int64

Black            51
Blue             39
White            19
Green            14
Gold              7
Grey              7
```

Let’s check the frequency of each value in every feature to check the difference in similar values.

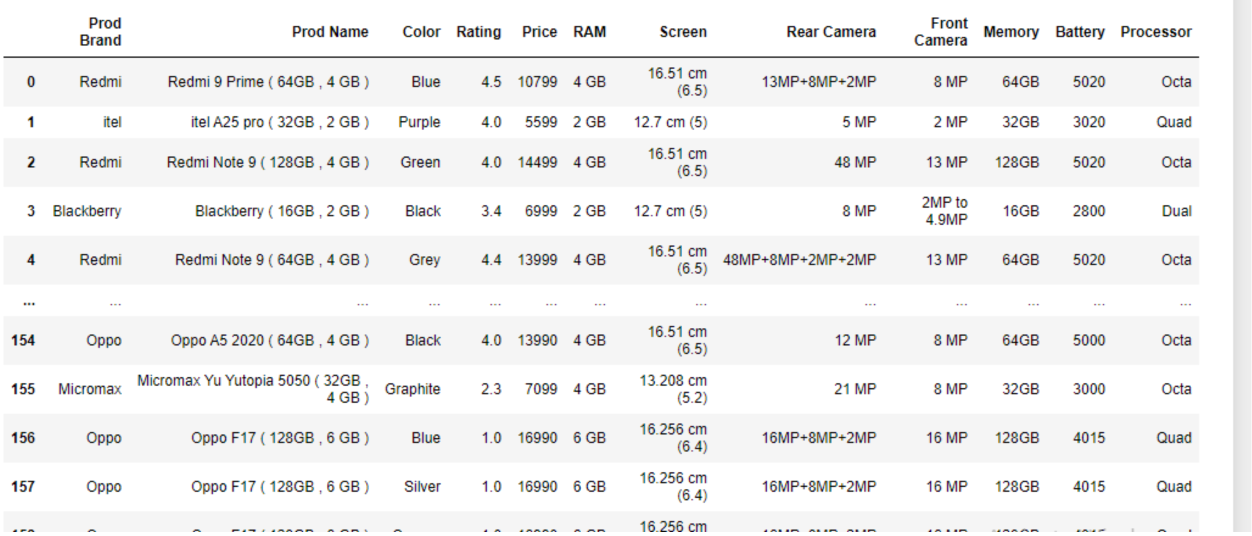


```
: battery = [i.split("m")[0].replace(" ", '') for i in df["Battery"]]
  df["Battery"]=battery
```



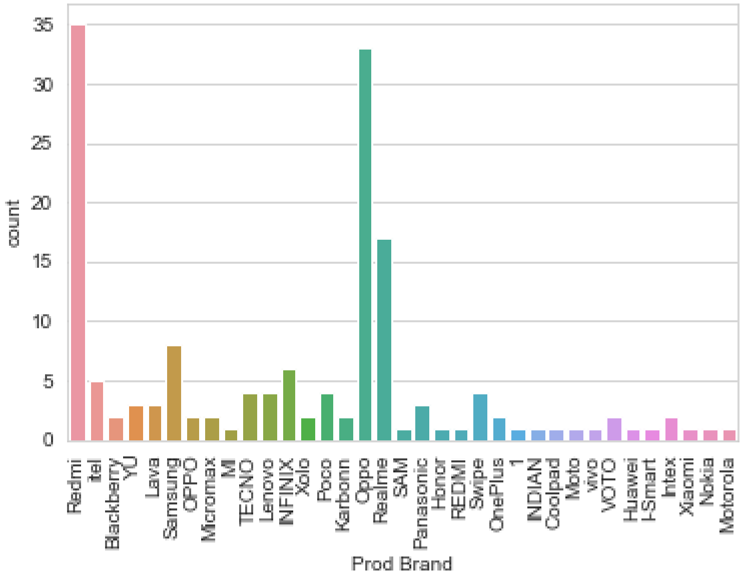
```
: df["Memory"].replace("64 GB","64GB",inplace=True)
```

After cleaning the data we will create a separate column as brand name from the product name attribute.

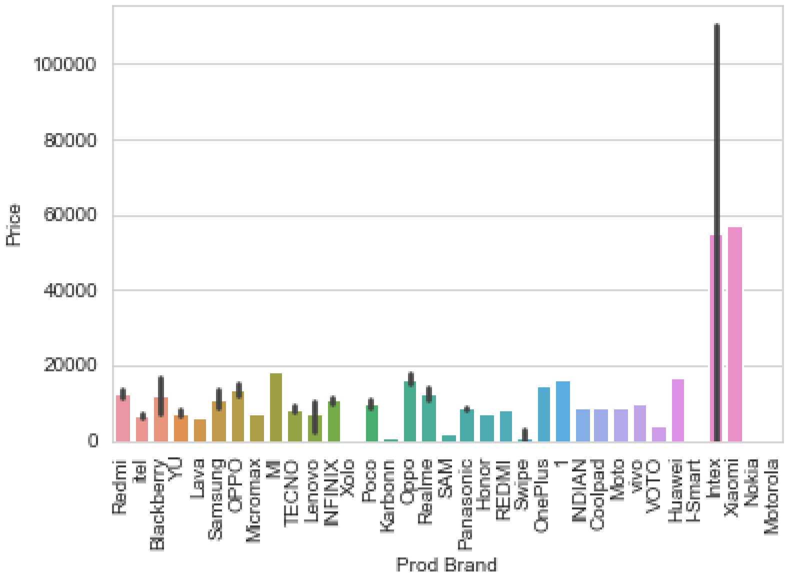


	Prod Brand	Prod Name	Color	Rating	Price	RAM	Screen	Rear Camera	Front Camera	Memory	Battery	Processor
0	Redmi	Redmi 9 Prime (64GB , 4 GB)	Blue	4.5	10799	4 GB	16.51 cm (6.5)	13MP+8MP+2MP	8 MP	64GB	5020	Octa
1	itel	itel A25 pro (32GB , 2 GB)	Purple	4.0	5599	2 GB	12.7 cm (5)	5 MP	2 MP	32GB	3020	Quad
2	Redmi	Redmi Note 9 (128GB , 4 GB)	Green	4.0	14499	4 GB	16.51 cm (6.5)	48 MP	13 MP	128GB	5020	Octa
3	Blackberry	Blackberry (16GB , 2 GB)	Black	3.4	6999	2 GB	12.7 cm (5)	8 MP	2MP to 4.9MP	16GB	2800	Dual
4	Redmi	Redmi Note 9 (64GB , 4 GB)	Grey	4.4	13999	4 GB	16.51 cm (6.5)	48MP+8MP+2MP+2MP	13 MP	64GB	5020	Octa
...	...	...	...	...	...	...	...	...	...	...	...	...
154	Oppo	Oppo A5 2020 (64GB , 4 GB)	Black	4.0	13990	4 GB	16.51 cm (6.5)	12 MP	8 MP	64GB	5000	Octa
155	Micromax	Micromax Yu Yutopia 5050 (32GB , 4 GB)	Graphite	2.3	7099	4 GB	13.208 cm (5.2)	21 MP	8 MP	32GB	3000	Octa
156	Oppo	Oppo F17 (128GB , 6 GB)	Blue	1.0	16990	6 GB	16.256 cm (6.4)	16MP+8MP+2MP	16 MP	128GB	4015	Quad
157	Oppo	Oppo F17 (128GB , 6 GB)	Silver	1.0	16990	6 GB	16.256 cm (6.4)	16MP+8MP+2MP	16 MP	128GB	4015	Quad
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16.256 cm	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

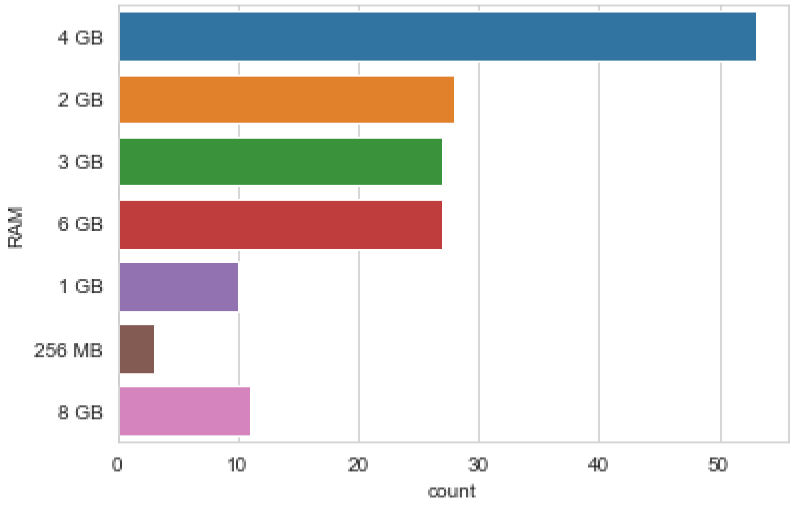
Now the data is ready for EDA.



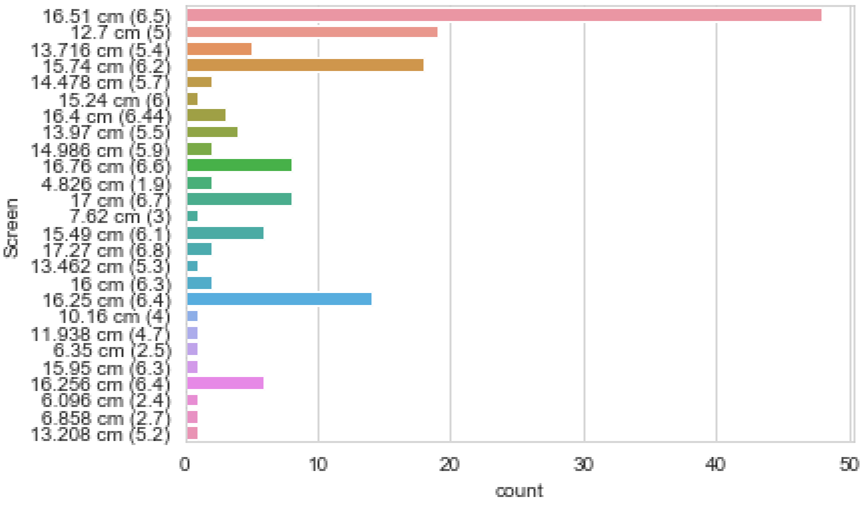
We see redmi oppo realme with most products in snapdeal. People buy more of these products.



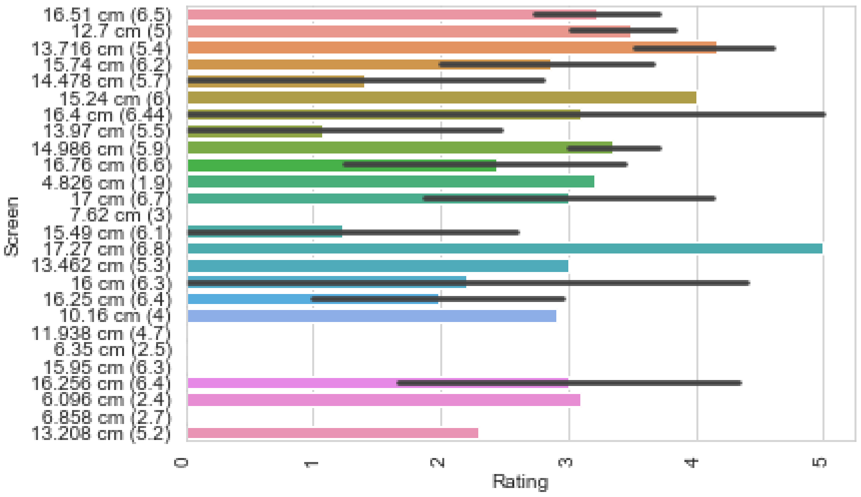
Most phones are below 20000 range in snapdeal.



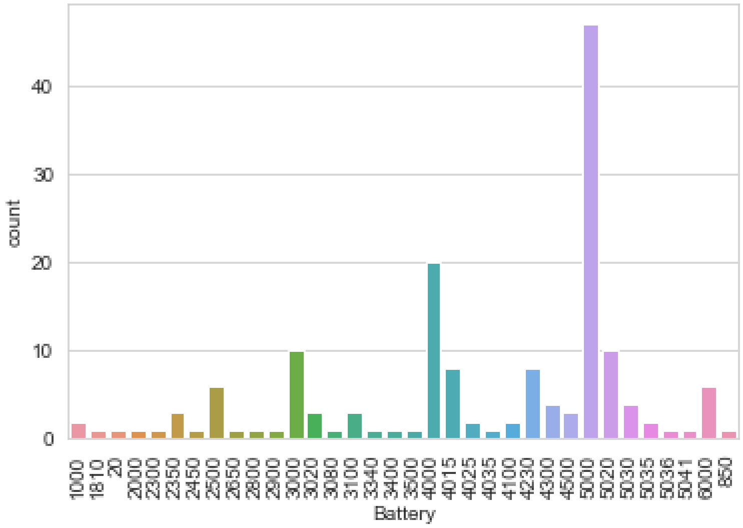
Most product are of 4 GB RAM in snapdeal. People buy more products with 4 GB RAM.



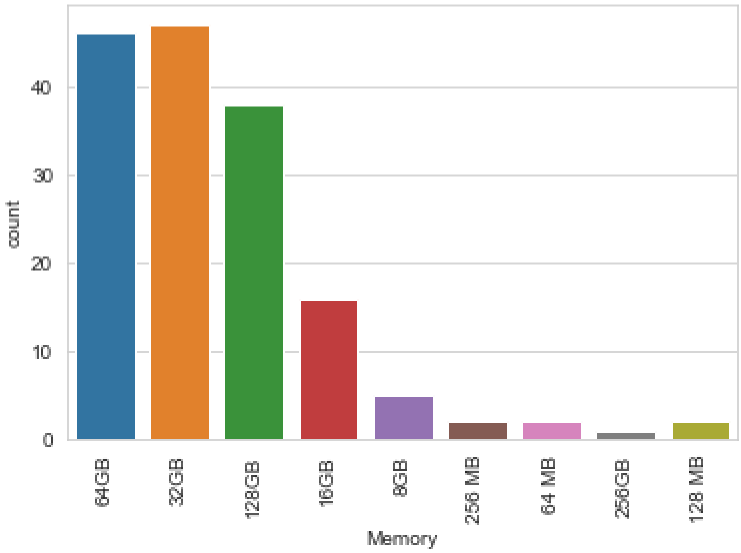
Maximum phones have screen of 16.51 cm range



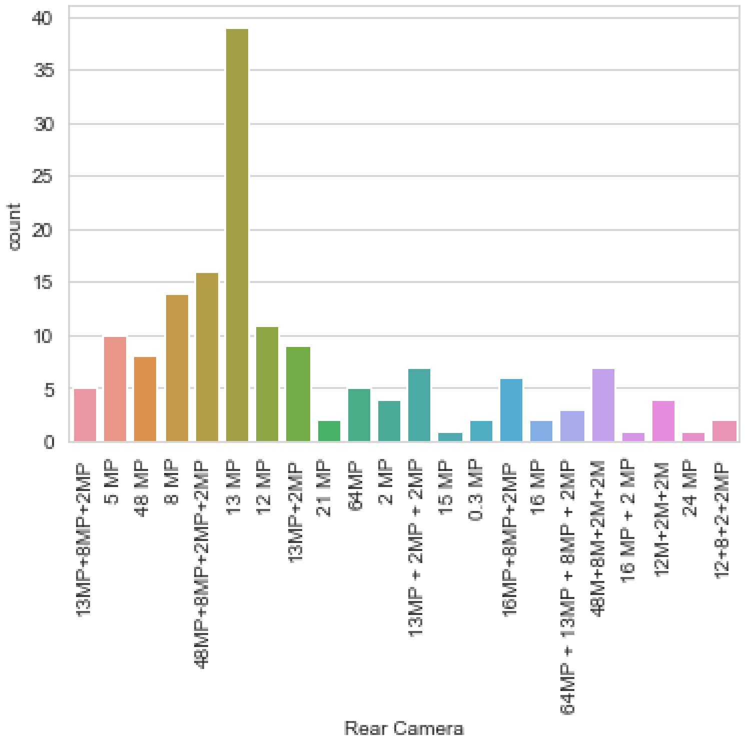
In above plot we see people rated the highest for the biggest screen size available in snapdeal.



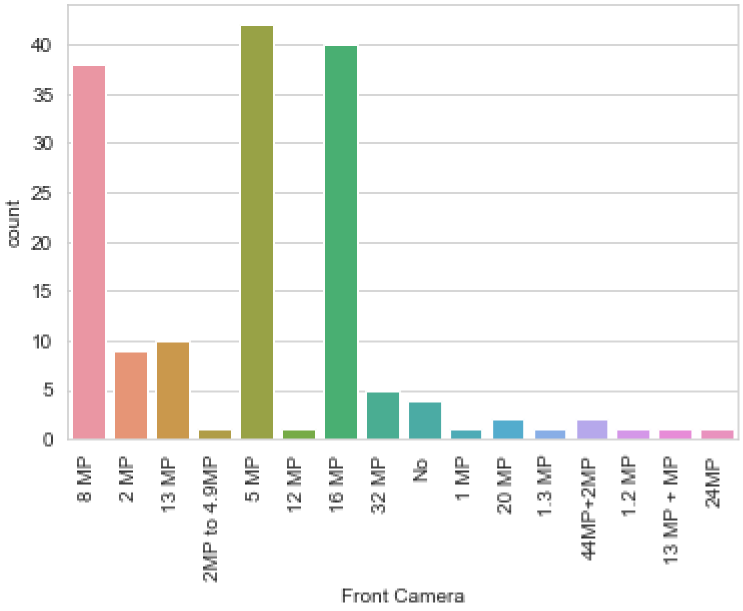
Most phones have battery capacity of 5000mAh.



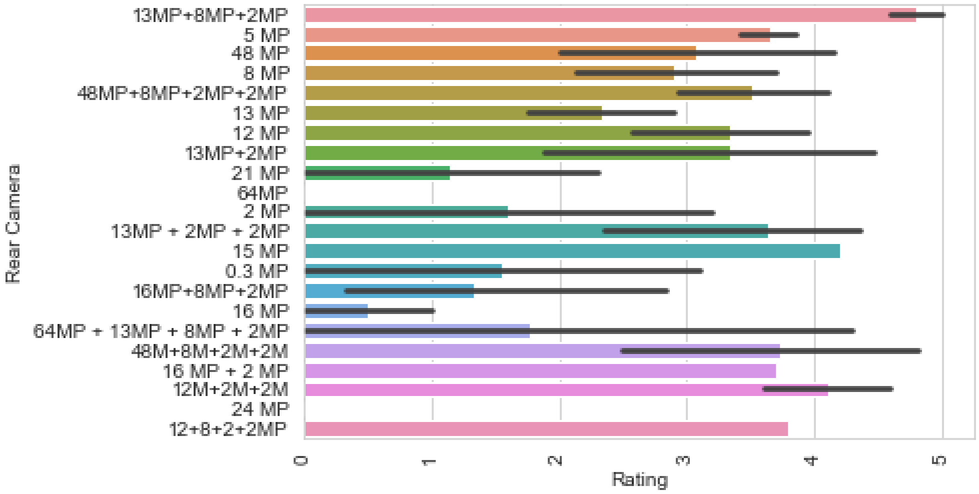
34 & 64 GB memory phones are in better price range.



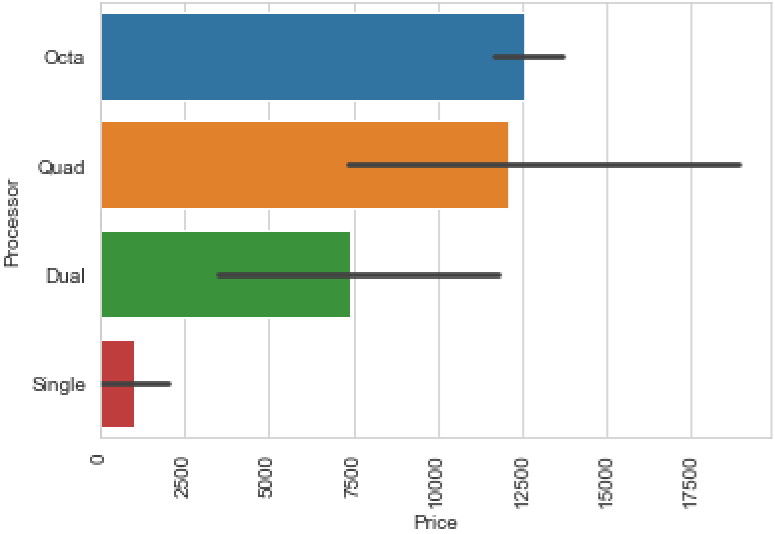
Maximum phones are of 13 MP rear camera.



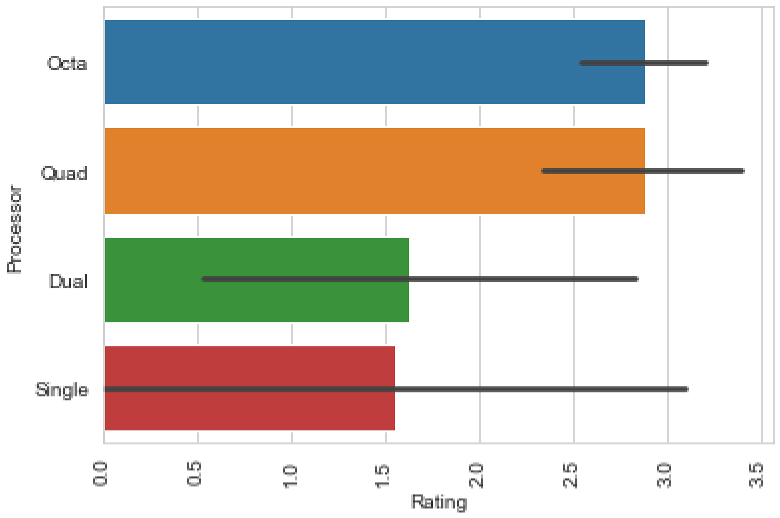
Maximum phones are of 5 , 16, or 8 MP front camera.



Phones with 13+8+2MP rear camera are highly rated.



As the processor power increases price also increases but few phones of quad core processors are highly rated.



Ratings are also high for higher processor phones. Few quad core processor phones are highly rated.

And so on.

From the above scrapped data one can find their own choice of mobile phones from snapdeal.

It was a good project full of learning's.

All files are saved as ipynb file (one for scraping and the other for EDA) , csv file (the scrapped data set), this pdf file as report.

THANK YOU