

*boAt Website Scraping*

A Project Submitted to the  
IT Vedant Institute, Thane.

Data Science & Data Analytics With AI



Python-Web-Scrapping Project

BY

Sachana Singh

Under the Guidance of

Mr. Sameer Warsolkar

# WEB SCRAPING PROJECT

---

## boAt Earbuds



# **USING PYTHON AND BEAUTIFULSOUP**

## DESCRIPTION

Earbuds web scrapping project involves extracting data from websites. This data can be used for various purposes, such as data analysis, research, reporting, or populating databases. We are going to use Ear buds to scrape site. To do so, we are going to use tools like Python, Requests, BeautifulSoup.

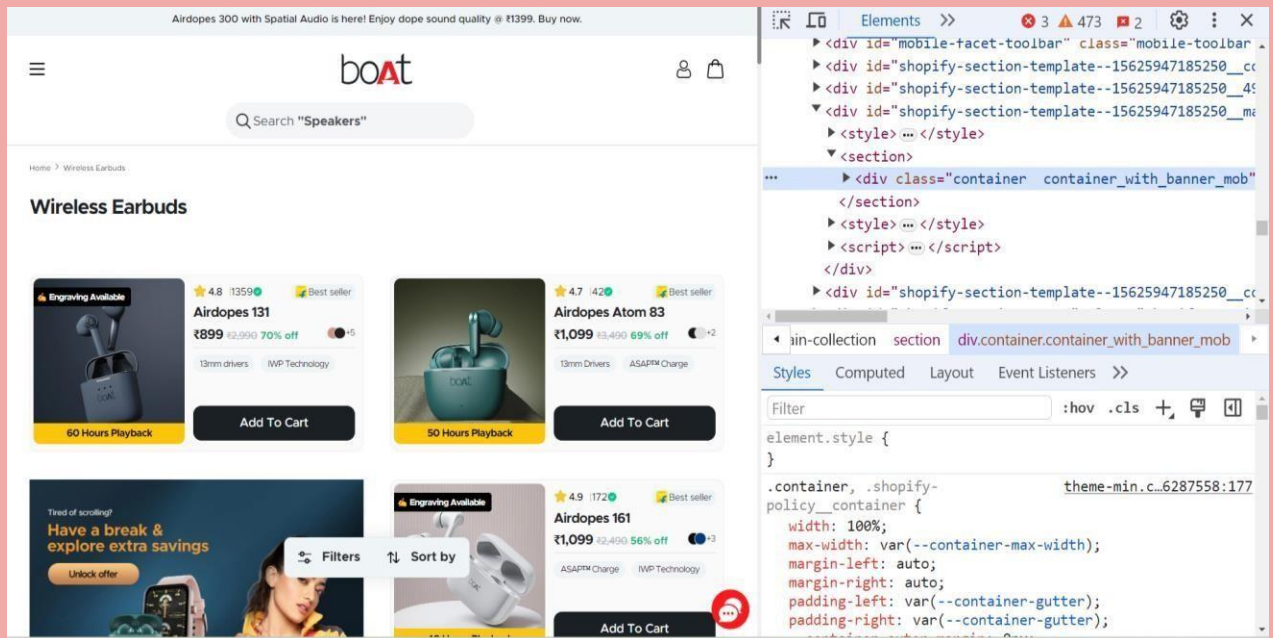
## OUTLINE

From this site, we are going to grab the following information:

- Ear buds Name
- Actual Price
- Discount
- Final Price
- Ratings
- Features
- Technology

## STEPS:

Choose the Website and Webpage URL  
Inspect the Website



## Installing the important libraries

Python has several web scrapping libraries. We will use the following libraries:

- Requests
- BeautifulSoup
- Pandas

Write the Python source code

Exporting the extracted data

## *boAt Earbuds Web Scraping Coding*

Accessing the Website using requests and BeautifulSoup Library , and Using html.parser

```
[ ]: #pip install bs4

[3]: import requests
    from bs4 import BeautifulSoup
    import pandas as pd

[4]: page = requests.get("https://www.boat-lifestyle.com/collections/true-wireless-earbuds/?utm_source=google")
    soup = BeautifulSoup(page.content, 'html.parser')
    soup

[4]: <!DOCTYPE html>
<html class="no-js" dir="ltr" draggable="false" lang="en">
<head>
<meta charset="utf-8"/>
```

```
[6]: print(soup.prettify())

<!DOCTYPE html>
<html class="no-js" dir="ltr" draggable="false" lang="en">
<head>
  <meta charset="utf-8"/>
  <meta content="width=device-width, initial-scale=1.0, height=device-height, minimum-scale=1.0, maximum-scale=6.0" name="viewport"/>
  <meta content="#ffffff" name="theme-color"/>
  <meta content="qad1e3gldwmxf61luncdygpvcduub" name="facebook-domain-verification"/>
  <link href="https://www.boat-lifestyle.com/collections/true-wireless-earbuds" rel="canonical"/>
  <link href="//www.boat-lifestyle.com/cdn/shop/files/32x32_256x256.png_32x32_2d0995d9-ec86-4c14-b928-71101777194c_96x96.png?v=1647426716" rel="shortcut icon" type="image/png"/>
  <title>
    Earbuds - Buy Wireless Earbuds Online in India | boAt
  </title>
```

Now accessing the Earbuds name.

```
File Edit View Run Kernel Settings Help Trusted
JupyterLab Python 3 (ipykernel)

[7]: Prod_Name = soup.select(".product-item-meta__title")
Prod_Name

[7]: [<a class="product-item-meta__title tile-title" href="/products/airdopes-131"> Airdopes 131 </a>,
<a class="product-item-meta__title tile-title" href="/products/boat-airdopes-atom-83-earbuds"> Airdope
s Atom 83 </a>,
<a class="product-item-meta__title tile-title" href="/products/airdopes-161"> Airdopes 161 </a>,
<a class="product-item-meta__title tile-title" href="/products/nirvana-ion-bluetooth-wireless-earbud
s"> Nirvana Ion </a>,
<a class="product-item-meta__title tile-title" href="/products/boat-airdopes-170-wireless-earbuds"> Ai
rdopes 170 </a>,
<a class="product-item-meta__title tile-title" href="/products/airdopes-141"> Airdopes 141 </a>,
<a class="product-item-meta__title tile-title" href="/products/boat-airdopes-131-pro-true-wireless-ear
buds"> Airdopes 131 PRO </a>,
<a class="product-item-meta__title tile-title" href="/products/airdopes-ace"> Airdopes Ace </a>,
<a class="product-item-meta__title tile-title" href="/products/boat-immortal-131-wireless-gaming-earbu
ds"> Immortal 131 </a>,
<a class="product-item-meta__title tile-title" href="/products/boat-airdopes-163-wireless-earbuds"> Ai
```

```
[8]: P_Name=[]
for i in Prod_Name:
    P_Name.append(i.get_text())
P_Name
#Len(P_Name)

[8]: [' Airdopes 131 ',
' Airdopes Atom 83 ',
' Airdopes 161 ',
' Nirvana Ion ',
' Airdopes 170 ',
' Airdopes 141 ',
' Airdopes 131 PRO ',
' Airdopes Ace ',
' Immortal 131 ',
' Airdopes 163 ',
' Airdopes 131 PRO ']
```

Accessing the Final price of every product.

```
File Edit View Run Kernel Settings Help Trusted
JupyterLab Python 3 (ipykernel)

[9]: Prod_Price = soup.select(".price.price--highlight.product-card-price")
Prod_Price

[9]: [<span class="price price--highlight product-card-price" data-price="89900">
<span class="visually-hidden">Sale price</span>₹899</span>,
<span class="price price--highlight product-card-price" data-price="109900">
<span class="visually-hidden">Sale price</span>₹1,099</span>,
<span class="price price--highlight product-card-price" data-price="109900">
<span class="visually-hidden">Sale price</span>₹1,099</span>,
<span class="price price--highlight product-card-price" data-price="199900">
<span class="visually-hidden">Sale price</span>₹1,999</span>,
<span class="price price--highlight product-card-price" data-price="159900">
<span class="visually-hidden">Sale price</span>₹1,599</span>,
<span class="price price--highlight product-card-price" data-price="119900">
<span class="visually-hidden">Sale price</span>₹1,299</span>,
<span class="price price--highlight product-card-price" data-price="109900">
<span class="visually-hidden">Sale price</span>₹1,099</span>,
<span class="price price--highlight product-card-price" data-price="129900">
<span class="visually-hidden">Sale price</span>₹1,299</span>]
```

```
File Edit View Run Kernel Settings Help Trusted
JupyterLab Python 3 (ipykernel)

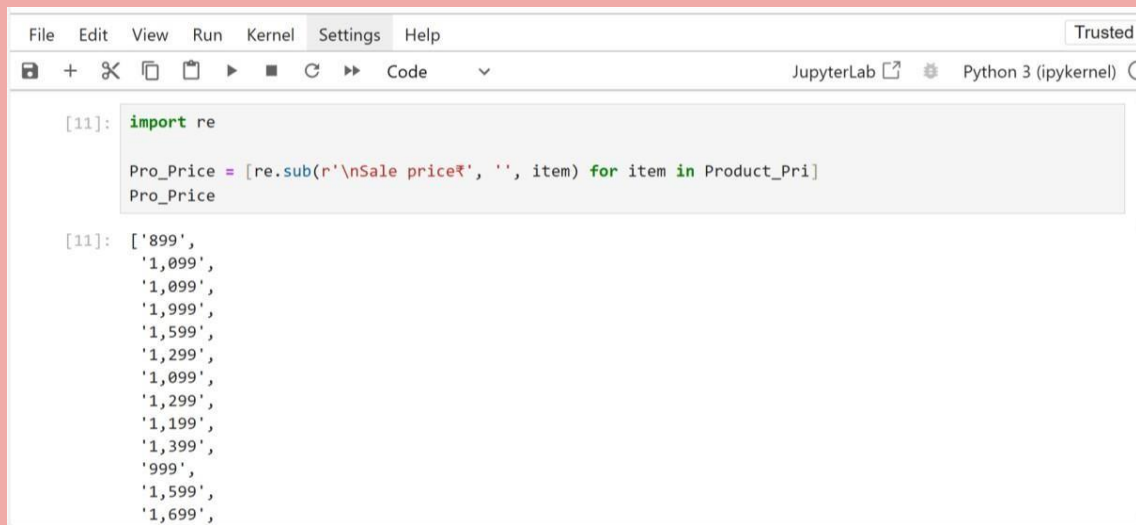
<span class="visually-hidden">Sale price</span>₹1,599</span>,
<span class="price price--highlight product-card-price" data-price="189900">
<span class="visually-hidden">Sale price</span>₹1,899</span>]

[10]: Product_Pri=[]
for i in Prod_Price :
    Product_Pri.append(i.get_text())
Product_Pri

[10]: ['\nSale price₹899',
'\nSale price₹1,099',
'\nSale price₹1,099',
'\nSale price₹1,099',
'\nSale price₹1,999',
'\nSale price₹1,599',
'\nSale price₹1,299',
'\nSale price₹1,099',
'\nSale price₹1,299',
'\nSale price₹1,199',
'\nSale price₹1,399',
```



## Using Regular Expression to extract specific data.



The image shows a JupyterLab interface with a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar. The code cell contains the following Python code:

```
[11]: import re

Pro_Price = [re.sub(r'\nSale price$', '', item) for item in Product_Pri]
Pro_Price
```

The output of the code cell is a list of price strings:

```
[11]: ['899',
      '1,099',
      '1,099',
      '1,999',
      '1,599',
      '1,299',
      '1,099',
      '1,299',
      '1,199',
      '1,399',
      '999',
      '1,599',
      '1,699',
```

Accessing the discount on every product.

```
[12]: Dis = soup.select(".m-0.pr-2.off")
      Dis

[12]: [<p class="m-0 pr-2 off">70% off</p>,
      <p class="m-0 pr-2 off">69% off</p>,
      <p class="m-0 pr-2 off">56% off</p>,
      <p class="m-0 pr-2 off">75% off</p>,
      <p class="m-0 pr-2 off">54% off</p>,
      <p class="m-0 pr-2 off">71% off</p>,
      <p class="m-0 pr-2 off">63% off</p>,
      <p class="m-0 pr-2 off">63% off</p>,
      <p class="m-0 pr-2 off">66% off</p>,
      <p class="m-0 pr-2 off">44% off</p>,
      <p class="m-0 pr-2 off">71% off</p>,
      <p class="m-0 pr-2 off">68% off</p>,
      <p class="m-0 pr-2 off">69% off</p>,
      <p class="m-0 pr-2 off">59% off</p>,
      <p class="m-0 pr-2 off">68% off</p>]
```

```
[13]: Discount=[]
      for i in Dis:
          Discount.append(i.get_text())
      Discount

[13]: ['70% off',
      '69% off',
      '56% off',
      '75% off',
      '54% off',
      '71% off',
      '63% off',
      '63% off',
      '66% off',
      '44% off',
      '71% off']
```

Using regular expression accordingly.

```
[28]: import re
      Discount_ = [re.sub(r'%\s*off','',item) for item in Discount]
      Discount_
      #Len(Discount_)

[28]: ['70',
      '71',
      '60',
      '75',
      '77',
      '76',
      '44',
      '63',
      '74',
      '63',
      '54',
      '59',
      '68',
      '66',
      '70',
      '47',
      '46']
```

Now accessing the **Ratings** on the Earbuds.

```
[15]: Rating_tags = soup.select(".rating__stars")
      Rating_tags
```

```
[15]: <div aria-label="4.84 out of 5.0 stars" class="rating_stars" role="img"><span>
  <svg height="12" viewBox="0 0 22 22" width="12" xmlns="http://www.w3.org/2000/svg">
    <g data-name="Group 334346" id="Group_334346" transform="translate(17150 -5434)">
      <rect data-name="Rectangle 114376" fill="transparent" height="22" id="Rectangle_114376" transform="tr
anslate(-17150 5434)" width="22"></rect>
      <g data-name="Group 334341" id="Group_334341" transform="translate(-17149 5435)">
        <path d="M1153.144,361.362a1.527,1.527,0,0,1,2.738,0l2.313,4.692,5.154.768a1.527,1.527,0,0,1,.848,2.6
1-3.729,3.683.868,5.179a1.526,1.526,0,0,1-2.216,1.61-4.606-2.424-4.607,2.424a1.526,1.526,0,0,1-2.216-
1.61.868-5.179-3.729-3.683a1.526,1.526,0,0,1,.848-2.615.154-.768Z" data-name="Path 332351" fill="#f4c7
30" fill-rule="evenodd" id="Path_332351" transform="translate(-1144.374 -360.511)"></path>
      </g>
    </g>
  </svg>
</span>
```

```
[16]: Rating_list=[]
      for i in Rating_tags:
          Rating_list.append(i.get_text())
```

Rating\_list

[illegible]

## Using Regular Expression.

```
[17]: import re

Rating_ = [re.sub(r'\s*\n+\s*', '', item) for item in Rating_list]
Rating_
#len(Rating_)
```

```
[17]: ['4.8',
       '4.7',
       '4.9',
       '5.0',
       '5.0',
       '4.9',
       '4.9',
       '4.6',
       '4.9',
       '4.8']
```

Now accessing some features of the product.

```
File Edit View Run Kernel Settings Help Trusted
JupyterLab Python 3 (ipykernel)

[18]: Feature_ = soup.select(".feature.feature-1")
      Feature_

[18]: [<span class="feature feature-1">13mm drivers </span>,
      <span class="feature feature-1">13mm Drivers </span>,
      <span class="feature feature-1">ASAP™ Charge </span>,
      <span class="feature feature-1">ENx™ Technology </span>,
      <span class="feature feature-1">ENx™ Technology </span>,
      <span class="feature feature-1">Clear Calling </span>,
      <span class="feature feature-1">Clear Calling </span>,
      <span class="feature feature-1">ASAP™ Charge </span>,
      <span class="feature feature-1">Up to 40 Hours Playback </span>,
      <span class="feature feature-1">boAt Immersive Sound </span>,
      <span class="feature feature-1">ENx™ Technology </span>,
      <span class="feature feature-1">ENx™ Tech </span>,
      <span class="feature feature-1">ENx™ Technology </span>,
      <span class="feature feature-1">BEAST™ Mode </span>,
      <span class="feature feature-1">Clear Calling </span>,
```

```
[19]: P_Feature=[]
      for i in Feature_:
          P_Feature.append(i.get_text())

      P_Feature
      #Len(P_Feature)

[19]: ['13mm drivers ',
      '13mm Drivers ',
      'ASAP™ Charge ',
      'ENx™ Technology ',
      'ENx™ Technology ',
      'Clear Calling ',
      'Clear Calling ',
      'ASAP™ Charge ',
      'Up to 40 Hours Playback ',
```

Now accessing the Technology of the product.

```
[18]: Tech = soup.select(".feature.feature-2")
Tech

[18]: [<span class="feature feature-2"> IWP Technology</span>,
<span class="feature feature-2"> ASAP™ Charge</span>,
<span class="feature feature-2"> IWP Technology</span>,
<span class="feature feature-2"> BEAST™ Mode </span>,
<span class="feature feature-2"> BEAST™ Mode </span>,
<span class="feature feature-2"> ASAP™ Charge</span>,
<span class="feature feature-2"> BEAST™ Mode</span>,
<span class="feature feature-2"> ENx™ Technology </span>,
<span class="feature feature-2"> IWP Technology</span>,
<span class="feature feature-2"> BEAST™ Mode </span>,
<span class="feature feature-2"> IWP™ Tech </span>,
<span class="feature feature-2"> ASAP™ Charge </span>,
<span class="feature feature-2"> BEAST™ Mode </span>,
<span class="feature feature-2"> 40 Hours Playback </span>,
<span class="feature feature-2"> ASAP™ Charge</span>]
```

```
[19]: Technology_=[]
for i in Tech:
    Technology_.append(i.get_text())

Technology_
#Len(Technology_)

[19]: [' IWP Technology',
' ASAP™ Charge',
' IWP Technology',
' BEAST™ Mode ',
' BEAST™ Mode ',
' ASAP™ Charge',
' BEAST™ Mode',
' ENx™ Technology ',
' IWP Technology',
' BEAST™ Mode ',
' IWP™ Tech ',
```

Accessing the actual and initial price of the product.

```
[22]: Act_price= soup.select(".price.price--compare.line-through")
      Act_price

[22]: [<span class="price price--compare line-through" data-compare-price="299000">
      <span class="visually-hidden">Regular price</span>₹2,990</span>,
      <span class="price price--compare line-through" data-compare-price="349000">
      <span class="visually-hidden">Regular price</span>₹3,490</span>,
      <span class="price price--compare line-through" data-compare-price="249000">
      <span class="visually-hidden">Regular price</span>₹2,490</span>,
      <span class="price price--compare line-through" data-compare-price="799000">
      <span class="visually-hidden">Regular price</span>₹7,990</span>,
      <span class="price price--compare line-through" data-compare-price="349000">
      <span class="visually-hidden">Regular price</span>₹3,490</span>,
      <span class="price price--compare line-through" data-compare-price="449000">
      <span class="visually-hidden">Regular price</span>₹4,490</span>,
      <span class="price price--compare line-through" data-compare-price="299000">
```

```
[23]: Actual_Pri_=[]
      for i in Act_price :
          Actual_Pri_.append(i.get_text())

      Actual_Pri_

[23]: ['\nRegular price₹2,990',
      '\nRegular price₹3,490',
      '\nRegular price₹2,490',
      '\nRegular price₹7,990',
      '\nRegular price₹3,490',
      '\nRegular price₹4,490',
      '\nRegular price₹2,990',
      '\nRegular price₹3,490',
      '\nRegular price₹3,490',
      '\nRegular price₹2,490',
```

```
[24]: import re

      Actual_Price = [re.sub(r'\nRegular price₹','', item) for item in Actual_Pri_]
      Actual_Price

[24]: ['2,990',
      '3,490',
      '2,490',
      '7,990',
      '3,490',
      '4,490',
      '2,990',
      '3,490',
      '3,490',
      '2,490',
      '3,490',
      '4,990',
      '5,490',
```

# Using Dataframe and pandas library.

File Edit View Run Kernel Settings Help Trusted

+

✂

📄

📄

▶

■

↺

▶▶

Code

▼

JupyterLab

Python 3 (ipykernel)

```
'3,499']

[39]: import pandas as pd
      Earbuds = pd.DataFrame({
          "Product Name":P_Name,
          "Actual Price": Actual_Price,
          "Discount":Discount_,
          "Final Price":Final_Pro_Price,
          "Ratings":Rating_,
          "Features":P_Feature,
          "More Features":Technology_
      })
      Earbuds
```

[23]:

	Product Name	Actual Price	Discount	Final Price	Ratings	Features	Technology
0	Airdopes 131	2,990	70%	899	4.8	13mm drivers	IWP Technology
1	Airdopes Atom 83	3,490	69%	1,099	4.7	13mm Drivers	ASAP™ Charge
2	Airdopes 161	2,490	56%	1,099	4.9	ASAP™ Charge	IWP Technology
3	Nirvana Ion	7,990	75%	1,999	5.0	ENx™ Technology	BEAST™ Mode
4	Airdopes 170	3,490	54%	1,599	5.0	ENx™ Technology	BEAST™ Mode
5	Airdopes 141	4,490	71%	1,299	4.9	Clear Calling	ASAP™ Charge
6	Airdopes 131 PRO	2,990	63%	1,099	4.9	Clear Calling	BEAST™ Mode
7	Airdopes Ace	3,490	63%	1,299	4.6	ASAP™ Charge	ENx™ Technology



## Exporting all the data into a CSV File.

```
[24]: Earbuds.to_csv("boAt_Report_Scraping.csv",index=False)
```

```
[25]: Read=pd.read_csv("boAt_Report_Scraping.csv")
Read
```

[25]:

	Product Name	Actual Price	Discount	Final Price	Ratings	Features	Technology
0	Airdopes 131	2,990	70%	899	4.8	13mm drivers	IWP Technology
1	Airdopes Atom 83	3,490	69%	1,099	4.7	13mm Drivers	ASAP™ Charge
2	Airdopes 161	2,490	56%	1,099	4.9	ASAP™ Charge	IWP Technology
3	Nirvana Ion	7,990	75%	1,999	5.0	ENx™ Technology	BEAST™ Mode
4	Airdopes 170	3,490	54%	1,599	5.0	ENx™ Technology	BEAST™ Mode
5	Airdopes 141	4,490	71%	1,299	4.9	Clear Calling	ASAP™ Charge

Open the file in the application and below is the Output for it.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Product Name	Actual Price	Discount	Final Price	Ratings	Features	Technology							
2	Airdopes 131	2,990	70%	899	4.8	13mm drivers	IWP Technology							
3	Airdopes Atom 83	3,490	69%	1,099	4.7	13mm Drivers	ASAPä,€ Charge							
4	Airdopes 161	2,490	56%	1,099	4.9	ASAPä,€ Charge	IWP Technology							
5	Nirvana Ion	7,990	75%	1,999	5	ENxä,€ Technology	BEASTä,€ Mode							
6	Airdopes 170	3,490	54%	1,599	5	ENxä,€ Technology	BEASTä,€ Mode							
7	Airdopes 141	4,490	71%	1,299	4.9	Clear Calling	ASAPä,€ Charge							
8	Airdopes 131 PRO	2,990	63%	1,099	4.9	Clear Calling	BEASTä,€ Mode							
9	Airdopes Ace	3,490	63%	1,299	4.6	ASAPä,€ Charge	ENxä,€ Technology							
10	Airdopes 163	2,490	44%	1,399	4.8	boAt Immersive Sound	IWP Technology							
11	Immortal 131	3,490	66%	1,199	4.9	Up to 40 Hours Playback	BEASTä,€ Mode							
12	Airdopes Ultra Plus	4,990	68%	1,599	4.8	ENxä,€ Tech	IWPä,€ Tech							
13	Airdopes Alpha	3,490	71%	999	4.9	ENxä,€ Technology	ASAPä,€ Charge							
14	Airdopes Hype	5,490	69%	1,699	4.5	ENxä,€ Technology	BEASTä,€ Mode							
15	Immortal 111	3,499	59%	1,449	5	BEASTä,€ Mode	40 Hours Playback							
16	Airdopes 148	4,490	68%	1,449	5	Clear Calling	ASAPä,€ Charge							
17	TRebel Airdopes 141	4,490	73%	1,199	5	Clear Calling	ASAPä,€ Charge							

# DATA VISUALIZATION

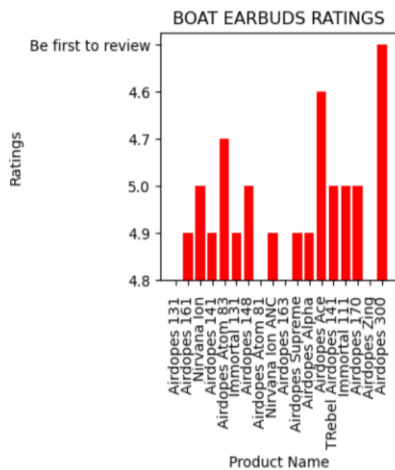
## 1) BAR PLOT

```
# DATA VISUALIZATION

[24]: import numpy as np
import pandas as pd
import seaborn as sns

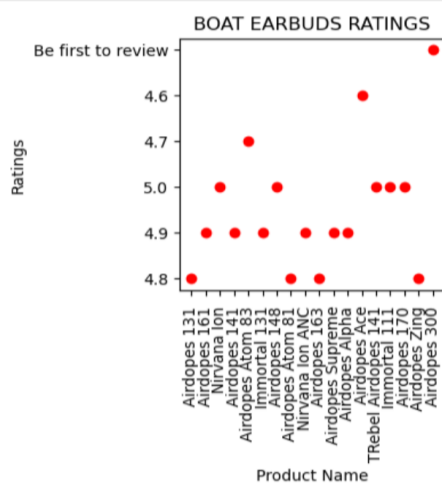
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

[47]: plt.figure(figsize=(3,3))
plt.bar(Earbuds['Product Name'], Earbuds['Ratings'], color='red')
plt.title('BOAT EARBUDS RATINGS')
plt.ylabel('Ratings')
plt.xlabel('Product Name')
plt.xticks(rotation=90)
plt.show()
```



## 2) SCATTER PLOT

```
[48]: plt.figure(figsize=(3,3))
plt.scatter(x=Earbuds['Product Name'], y=Earbuds['Ratings'], color='red')
plt.title('BOAT EARBUDS RATINGS')
plt.ylabel('Ratings')
plt.xlabel('Product Name')
plt.xticks(rotation=90)
plt.show()
```



## CONCLUSION

*In conclusion, the web scraping project focused on boAt Earbuds has successfully achieved its objectives. The omprehensive data retrieval provided detailed specifications, product name, price, ratings , discounts and special features of the product .*



*Thank You*