I.T. VEDANT

Decode your dreams

Name : Rigved Jagdish Tupsakhare

Project Name : Web Scraping Tata Motor Cars From ZigWheels.

Project Guide : Prof. Sameer Warsolkar

Web Scraping Tata Motor Cars From ZigWheels.Com

1.Aim:

To gather comprehensive information on Tata Motors cars from ZigWheels website through web scraping, providing valuable insights into the specifications, features, and pricing.

2.Objectives:

1.Data Collection:

- Extract detailed information on Tata Motors car models available on ZigWheels, including specifications such as engine type, power, torque, fuel efficiency, etc.
- Scrape data related to car features, such as safety features, entertainment options, interior, and exterior design.

2.Pricing Analysis:

- Retrieve pricing information for each Tata Motors car model and variant.
- Compare prices across different variants and models to identify trends and variations.

3.Market Trends:

- Analyze the popularity and demand for Tata Motors cars by scraping user reviews and ratings.
- Identify key features and specifications that are frequently mentioned in positive or negative reviews.

4.Competitor Analysis:

- ➢ Compare Tata Motors car specifications and pricing with competitors in the market.
- ➢ Evaluate how Tata Motors models stand out or compete within their respective segments.

5.Data Quality Assurance:

- ➢ Implement mechanisms to ensure data accuracy and completeness during the scraping process.
- ➢ Address potential challenges such as changes in website structure that might affect the scraping workflow.

3. OUTLINE:

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

1. Car_Names
2. Engine|Mileage|fueltype
3. Car_Prices
4. Ratings
5. Total_Reviews

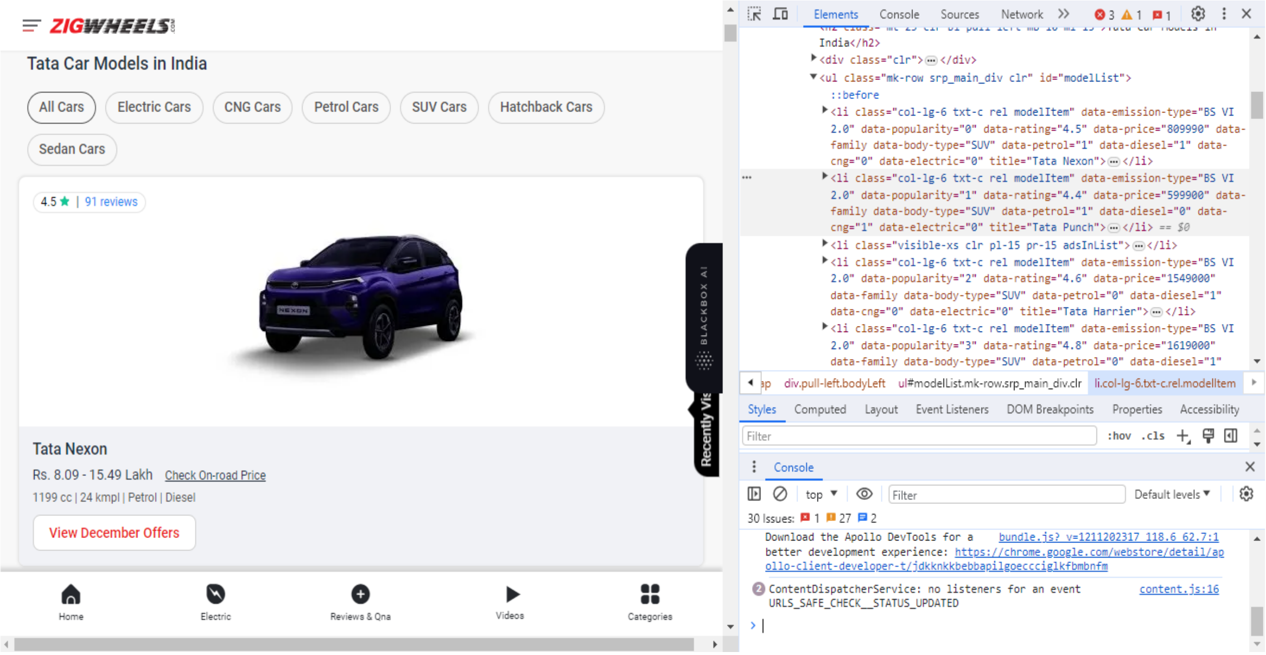
4. STEPS:

Choose the Website and Webpage URL:

➢ The first step is to select the website you want to scrape. We will try to extract data of Tata cars from the ZigWheels.com.

1. Inspect the website:

➢ Now the next step is to understand the website structure. Understand what the attributes of the elements that are of your interest are. Right click on the website to select “Inspect”. This will open HTML code. Use the inspector tool to see the name of all the elements to use in the code.



```
India</h2>
▶<div class="clr">⋯</div>
▼<ul class="mk-row srp_main_div clr" id="modelList">
   ::before
  ▶<li class="col-lg-6 txt-c rel modelItem" data-emission-type="BS VI 2.0" data-popularity="0" data-rating="4.5" data-price="809990" data-family data-body-type="SUV" data-petrol="1" data-diesel="1" data-cng="0" data-electric="0" title="Tata Nexon">⋯</li>
  ▶<li class="col-lg-6 txt-c rel modelItem" data-emission-type="BS VI 2.0" data-popularity="1" data-rating="4.4" data-price="599900" data-family data-body-type="SUV" data-petrol="1" data-diesel="0" data-cng="1" data-electric="0" title="Tata Punch">⋯</li> == $0
  ▶<li class="visible-xs clr pl-15 pr-15 adsInList">⋯</li>
  ▶<li class="col-lg-6 txt-c rel modelItem" data-emission-type="BS VI 2.0" data-popularity="2" data-rating="4.6" data-price="1549000" data-family data-body-type="SUV" data-petrol="0" data-diesel="1" data-cng="0" data-electric="0" title="Tata Harrier">⋯</li>
  ▶<li class="col-lg-6 txt-c rel modelItem" data-emission-type="BS VI 2.0" data-popularity="3" data-rating="4.8" data-price="1619000" data-family data-body-type="SUV" data-petrol="0" data-diesel="1"

ap  div.pull-left.bodyLeft  ul#modelList.mk-row.srp_main_div.clr  li.col-lg-6.txt-c.rel.modelItem
```

```
Download the Apollo DevTools for a          bundle.js? v=1211202317 118.6 62.7:1
better development experience: https://chrome.google.com/webstore/detail/apollo-client-developer-t/jdkknkkbebbapilgoeccciglkfbmbnfm
2 ContentDispatcherService: no listeners for an event          content.js:16
URLS_SAFE_CHECK__STATUS_UPDATED
>
```

2. Installing the important libraries:

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

➢ Requests – for making HTTP requests to website

➢ BeautifulSoup – for parsing the HTML code

➢ Pandas – for storing the scraped data in data frame

3. Write the Python source code:

We’ll write the main python code. The code will perform the following steps:

➢ Using requests to send an HTTP GET requests

➢ Using BeautifulSoup to parse the HTML code

➢ Extracting the required data from the HTML code

➢ Store the information in a pandas DataFrame

4. Exporting the extracted data:

We'll export the data as a CSV file. We will use the pandas library. We'll use the pandas library.

5. Benefits:

➢ Access to valuable data for analysis or research.

➢ Automation of data collection, saving time and effort.

➢ Stay up to date with change changes on the target websites.

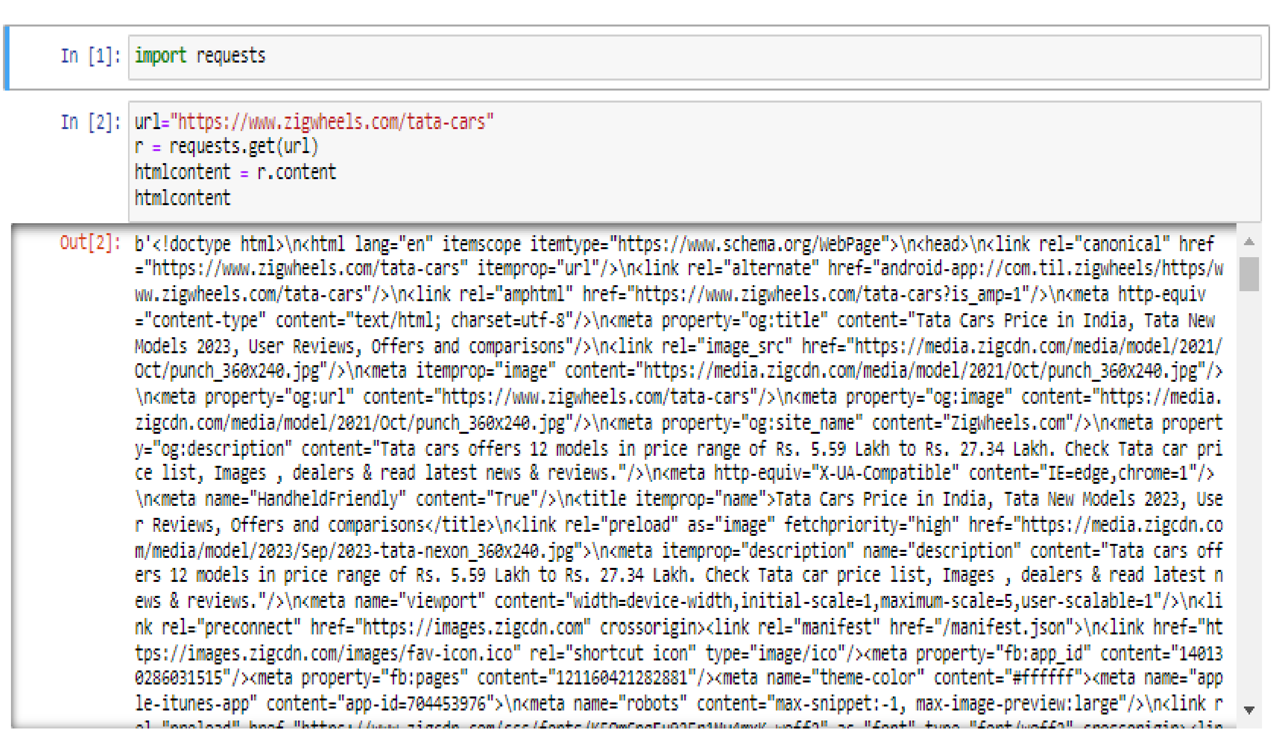
6. Risk:

➢ Legal issues related to web scraping.

➢ Technical challenges due to website changes

5. ZigWheels web scraping coding command steps

1. Accessing Tata Motors on ZigWheels website:

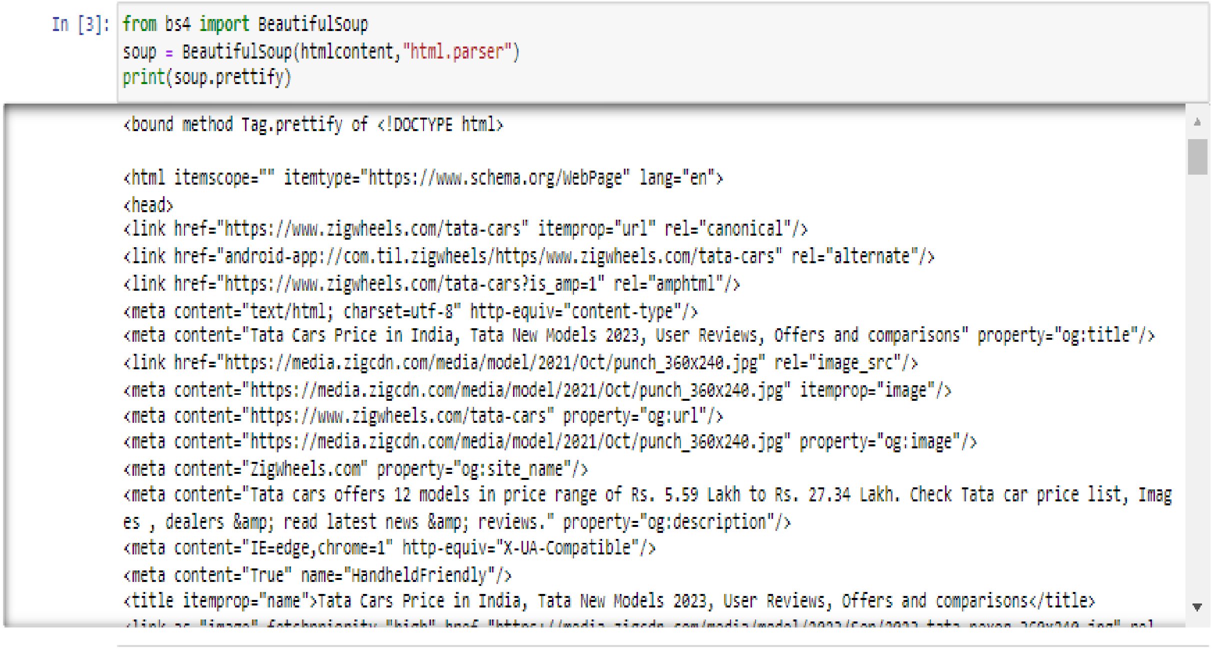


In [1]:
```
import requests
```

In [2]:
```
url="https://www.zigwheels.com/tata-cars"
r = requests.get(url)
htmlcontent = r.content
htmlcontent
```

Out[2]:
```
b'<!doctype html>\n<html lang="en" itemscope itemtype="https://www.schema.org/WebPage">\n<head>\n<link rel="canonical" href
="https://www.zigwheels.com/tata-cars" itemprop="url"/>\n<link rel="alternate" href="android-app://com.til.zigwheels/https/w
ww.zigwheels.com/tata-cars"/>\n<link rel="amphtml" href="https://www.zigwheels.com/tata-cars?is_amp=1"/>\n<meta http-equiv
="content-type" content="text/html; charset=utf-8"/>\n<meta property="og:title" content="Tata Cars Price in India, Tata New
Models 2023, User Reviews, Offers and comparisons"/>\n<link rel="image_src" href="https://media.zigcdn.com/media/model/2021/
Oct/punch_360x240.jpg"/>\n<meta itemprop="image" content="https://media.zigcdn.com/media/model/2021/Oct/punch_360x240.jpg"/>
\n<meta property="og:url" content="https://www.zigwheels.com/tata-cars"/>\n<meta property="og:image" content="https://media.
zigcdn.com/media/model/2021/Oct/punch_360x240.jpg"/>\n<meta property="og:site_name" content="ZigWheels.com"/>\n<meta propert
y="og:description" content="Tata cars offers 12 models in price range of Rs. 5.59 Lakh to Rs. 27.34 Lakh. Check Tata car pri
ce list, Images , dealers & read latest news & reviews."/>\n<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"/>
\n<meta name="HandheldFriendly" content="True"/>\n<title itemprop="name">Tata Cars Price in India, Tata New Models 2023, Use
r Reviews, Offers and comparisons</title>\n<link rel="preload" as="image" fetchpriority="high" href="https://media.zigcdn.co
m/media/model/2023/Sep/2023-tata-nexon_360x240.jpg">\n<meta itemprop="description" name="description" content="Tata cars off
ers 12 models in price range of Rs. 5.59 Lakh to Rs. 27.34 Lakh. Check Tata car price list, Images , dealers & read latest n
ews & reviews."/>\n<meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=5,user-scalable=1"/>\n<li
nk rel="preconnect" href="https://images.zigcdn.com" crossorigin><link rel="manifest" href="/manifest.json">\n<link href="ht
tps://images.zigcdn.com/images/fav-icon.ico" rel="shortcut icon" type="image/ico"/><meta property="fb:app_id" content="14013
0286031515"/><meta property="fb:pages" content="121160421282881"/><meta name="theme-color" content="#ffffff"><meta name="app
le-itunes-app" content="app-id=704453976">\n<meta name="robots" content="max-snippet:-1, max-image-preview:large"/>\n<link r
```

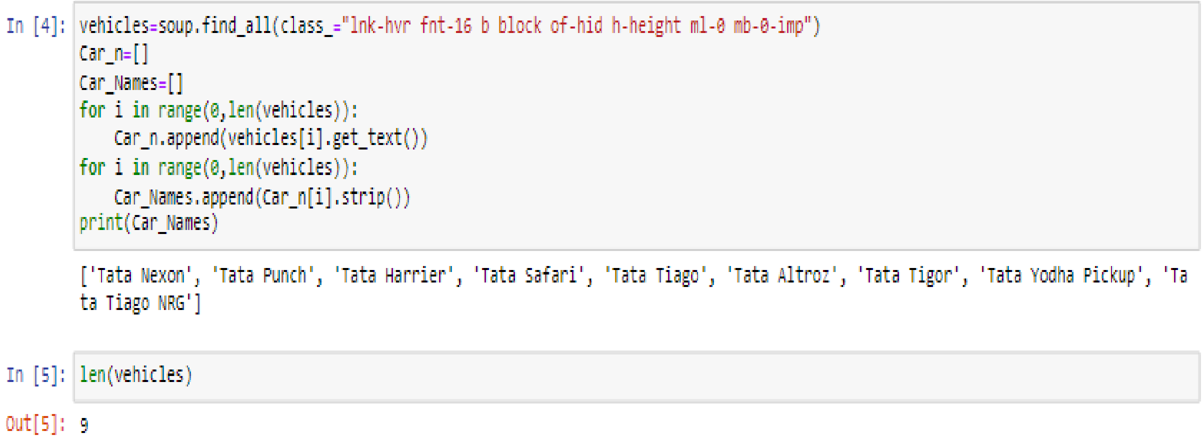
2. Using BeautifulSoup :



In [3]:
```
from bs4 import BeautifulSoup
soup = BeautifulSoup(htmlcontent,"html.parser")
print(soup.prettify)
```

```
<bound method Tag.prettify of <!DOCTYPE html>

<html itemscope="" itemtype="https://www.schema.org/WebPage" lang="en">
<head>
<link href="https://www.zigwheels.com/tata-cars" itemprop="url" rel="canonical"/>
<link href="android-app://com.til.zigwheels/https/www.zigwheels.com/tata-cars" rel="alternate"/>
<link href="https://www.zigwheels.com/tata-cars?is_amp=1" rel="amphtml"/>
<meta content="text/html; charset=utf-8" http-equiv="content-type"/>
<meta content="Tata Cars Price in India, Tata New Models 2023, User Reviews, Offers and comparisons" property="og:title"/>
<link href="https://media.zigcdn.com/media/model/2021/Oct/punch_360x240.jpg" rel="image_src"/>
<meta content="https://media.zigcdn.com/media/model/2021/Oct/punch_360x240.jpg" itemprop="image"/>
<meta content="https://www.zigwheels.com/tata-cars" property="og:url"/>
<meta content="https://media.zigcdn.com/media/model/2021/Oct/punch_360x240.jpg" property="og:image"/>
<meta content="ZigWheels.com" property="og:site_name"/>
<meta content="Tata cars offers 12 models in price range of Rs. 5.59 Lakh to Rs. 27.34 Lakh. Check Tata car price list, Imag
es , dealers &amp; read latest news &amp; reviews." property="og:description"/>
<meta content="IE=edge,chrome=1" http-equiv="X-UA-Compatible"/>
<meta content="True" name="HandheldFriendly"/>
<title itemprop="name">Tata Cars Price in India, Tata New Models 2023, User Reviews, Offers and comparisons</title>
```

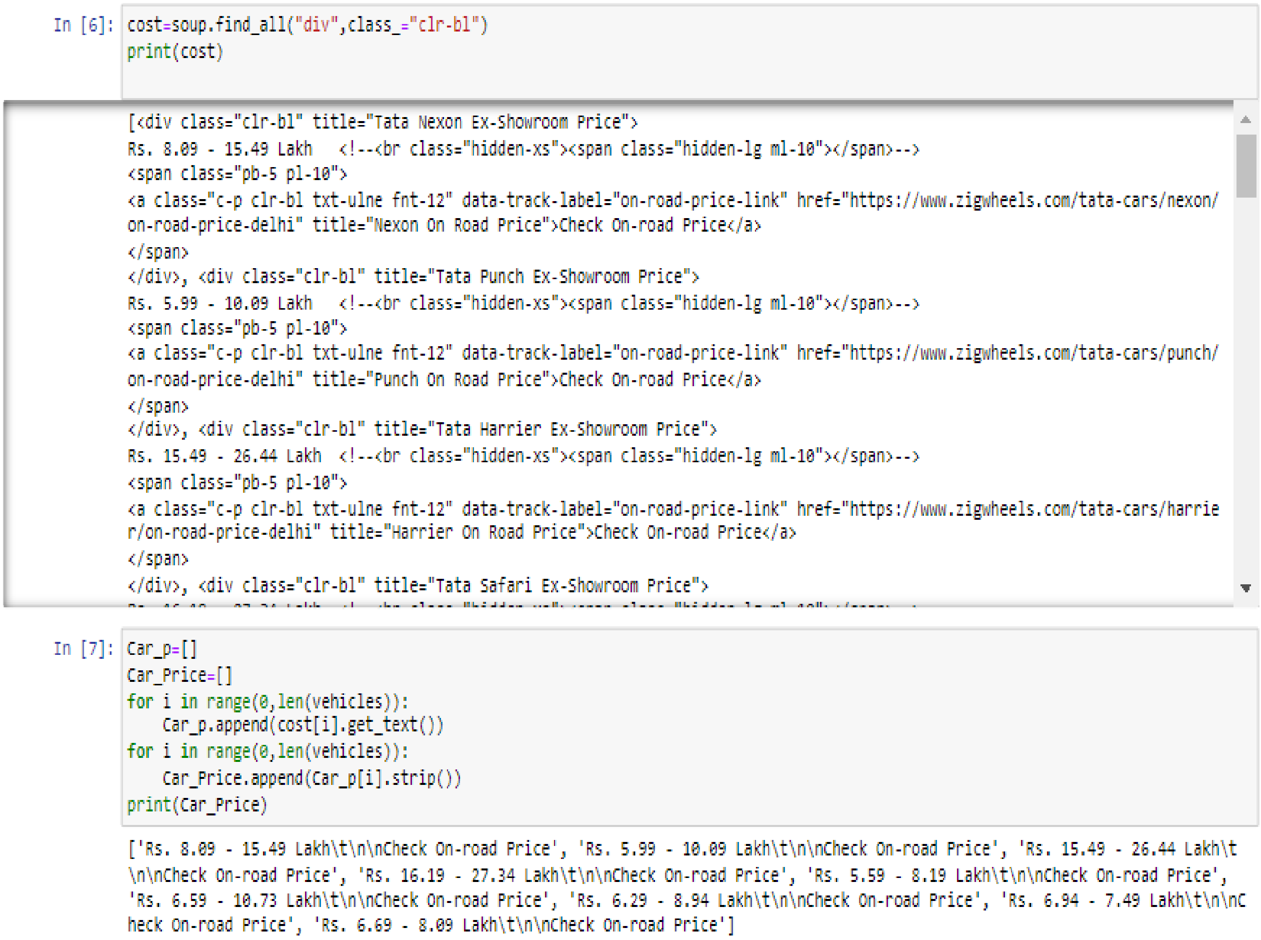
3. Accessing the Car_Names:



```
In [4]: vehicles=soup.find_all(class_="lnk-hvr fnt-16 b block of-hid h-height ml-0 mb-0-imp")
        Car_n=[]
        Car_Names=[]
        for i in range(0,len(vehicles)):
            Car_n.append(vehicles[i].get_text())
        for i in range(0,len(vehicles)):
            Car_Names.append(Car_n[i].strip())
        print(Car_Names)

        ['Tata Nexon', 'Tata Punch', 'Tata Harrier', 'Tata Safari', 'Tata Tiago', 'Tata Altroz', 'Tata Tigor', 'Tata Yodha Pickup', 'Ta
        ta Tiago NRG']

In [5]: len(vehicles)

Out[5]: 9
```

---

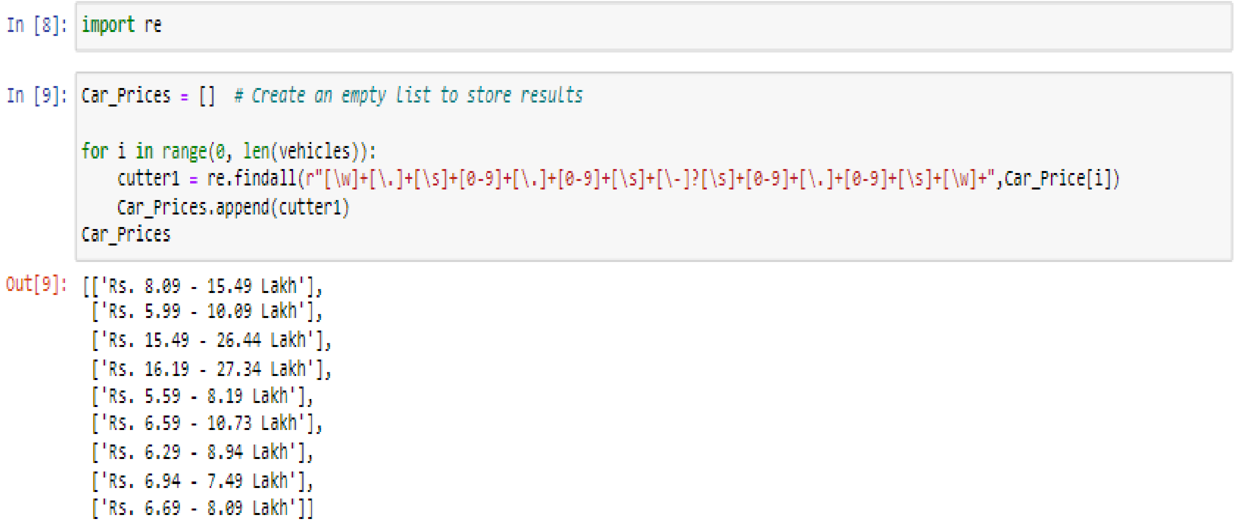
4. Accessing the Car_Prices:



```
In [6]: cost=soup.find_all("div",class_="clr-bl")
        print(cost)

        [<div class="clr-bl" title="Tata Nexon Ex-Showroom Price">
        Rs. 8.09 - 15.49 Lakh   <!--<br class="hidden-xs"><span class="hidden-lg ml-10"></span>-->
        <span class="pb-5 pl-10">
        <a class="c-p clr-bl txt-ulne fnt-12" data-track-label="on-road-price-link" href="https://www.zigwheels.com/tata-cars/nexon/
        on-road-price-delhi" title="Nexon On Road Price">Check On-road Price</a>
        </span>
        </div>, <div class="clr-bl" title="Tata Punch Ex-Showroom Price">
        Rs. 5.99 - 10.09 Lakh   <!--<br class="hidden-xs"><span class="hidden-lg ml-10"></span>-->
        <span class="pb-5 pl-10">
        <a class="c-p clr-bl txt-ulne fnt-12" data-track-label="on-road-price-link" href="https://www.zigwheels.com/tata-cars/punch/
        on-road-price-delhi" title="Punch On Road Price">Check On-road Price</a>
        </span>
        </div>, <div class="clr-bl" title="Tata Harrier Ex-Showroom Price">
        Rs. 15.49 - 26.44 Lakh  <!--<br class="hidden-xs"><span class="hidden-lg ml-10"></span>-->
        <span class="pb-5 pl-10">
        <a class="c-p clr-bl txt-ulne fnt-12" data-track-label="on-road-price-link" href="https://www.zigwheels.com/tata-cars/harrie
        r/on-road-price-delhi" title="Harrier On Road Price">Check On-road Price</a>
        </span>
        </div>, <div class="clr-bl" title="Tata Safari Ex-Showroom Price">

In [7]: Car_p=[]
        Car_Price=[]
        for i in range(0,len(vehicles)):
            Car_p.append(cost[i].get_text())
        for i in range(0,len(vehicles)):
            Car_Price.append(Car_p[i].strip())
        print(Car_Price)

        ['Rs. 8.09 - 15.49 Lakh\t\n\nCheck On-road Price', 'Rs. 5.99 - 10.09 Lakh\t\n\nCheck On-road Price', 'Rs. 15.49 - 26.44 Lakh\t
        \n\nCheck On-road Price', 'Rs. 16.19 - 27.34 Lakh\t\n\nCheck On-road Price', 'Rs. 5.59 - 8.19 Lakh\t\n\nCheck On-road Price',
        'Rs. 6.59 - 10.73 Lakh\t\n\nCheck On-road Price', 'Rs. 6.29 - 8.94 Lakh\t\n\nCheck On-road Price', 'Rs. 6.94 - 7.49 Lakh\t\n\nC
        heck On-road Price', 'Rs. 6.69 - 8.09 Lakh\t\n\nCheck On-road Price']
```

---

5. Using RegEx to get the required text(Car_Prices) from the list:

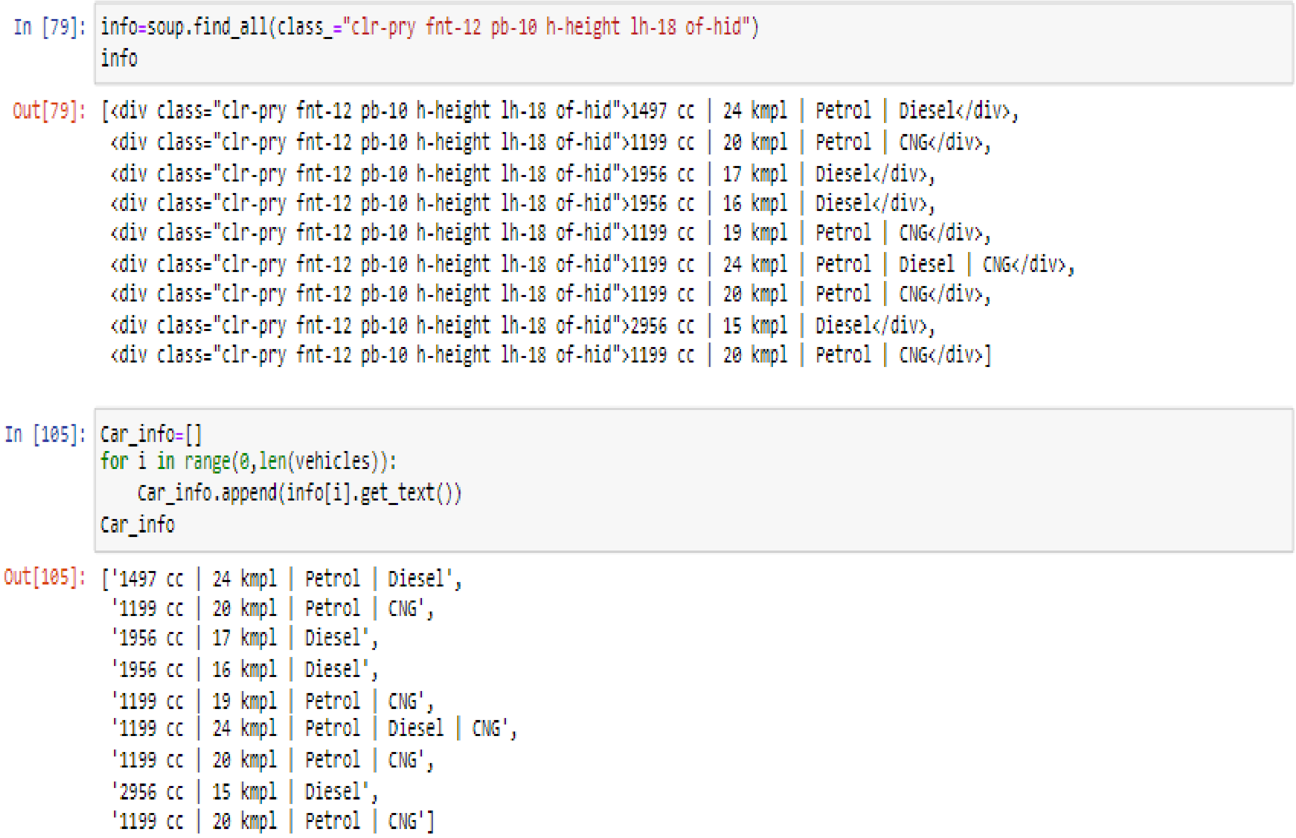


```
In [8]: import re
```

```
In [9]: Car_Prices = []  # Create an empty list to store results

        for i in range(0, len(vehicles)):
            cutter1 = re.findall(r"[\w]+[\.]+[\s]+[0-9]+[\.]+[0-9]+[\s]+[\-]?[\s]+[0-9]+[\.]+[0-9]+[\s]+[\w]+",Car_Price[i])
            Car_Prices.append(cutter1)
        Car_Prices
```

```
Out[9]: [['Rs. 8.09 - 15.49 Lakh'],
         ['Rs. 5.99 - 10.09 Lakh'],
         ['Rs. 15.49 - 26.44 Lakh'],
         ['Rs. 16.19 - 27.34 Lakh'],
         ['Rs. 5.59 - 8.19 Lakh'],
         ['Rs. 6.59 - 10.73 Lakh'],
         ['Rs. 6.29 - 8.94 Lakh'],
         ['Rs. 6.94 - 7.49 Lakh'],
         ['Rs. 6.69 - 8.09 Lakh']]
```

6. Accessing Engine|Mileage|fueltype:



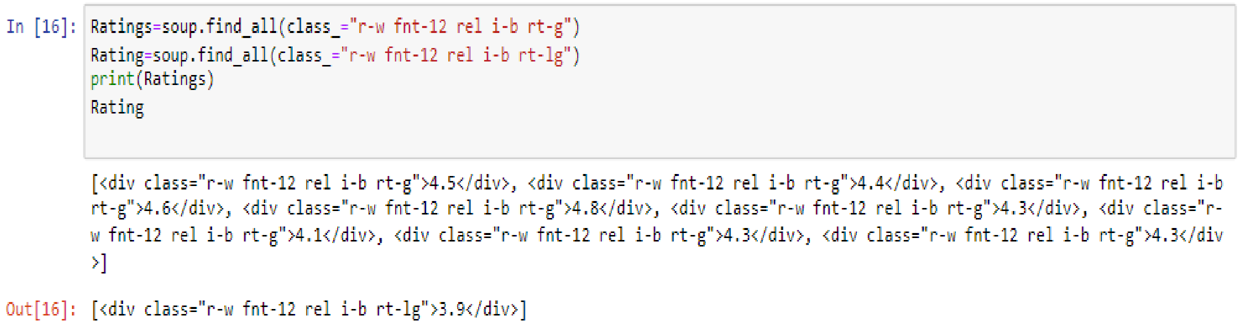
```
In [79]: info=soup.find_all(class_="clr-pry fnt-12 pb-10 h-height lh-18 of-hid")
         info
```

```
Out[79]: [<div class="clr-pry fnt-12 pb-10 h-height lh-18 of-hid">1497 cc | 24 kmpl | Petrol | Diesel</div>,
          <div class="clr-pry fnt-12 pb-10 h-height lh-18 of-hid">1199 cc | 20 kmpl | Petrol | CNG</div>,
          <div class="clr-pry fnt-12 pb-10 h-height lh-18 of-hid">1956 cc | 17 kmpl | Diesel</div>,
          <div class="clr-pry fnt-12 pb-10 h-height lh-18 of-hid">1956 cc | 16 kmpl | Diesel</div>,
          <div class="clr-pry fnt-12 pb-10 h-height lh-18 of-hid">1199 cc | 19 kmpl | Petrol | CNG</div>,
          <div class="clr-pry fnt-12 pb-10 h-height lh-18 of-hid">1199 cc | 24 kmpl | Petrol | Diesel | CNG</div>,
          <div class="clr-pry fnt-12 pb-10 h-height lh-18 of-hid">1199 cc | 20 kmpl | Petrol | CNG</div>,
          <div class="clr-pry fnt-12 pb-10 h-height lh-18 of-hid">2956 cc | 15 kmpl | Diesel</div>,
          <div class="clr-pry fnt-12 pb-10 h-height lh-18 of-hid">1199 cc | 20 kmpl | Petrol | CNG</div>]
```

```
In [105]: Car_info=[]
          for i in range(0,len(vehicles)):
              Car_info.append(info[i].get_text())
          Car_info
```

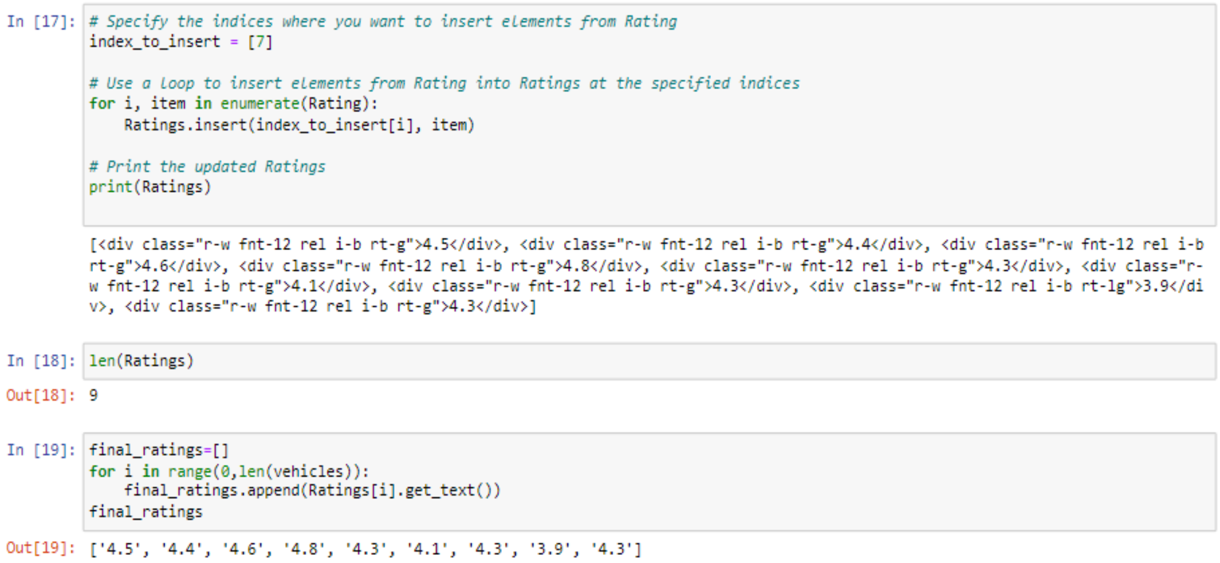
```
Out[105]: ['1497 cc | 24 kmpl | Petrol | Diesel',
           '1199 cc | 20 kmpl | Petrol | CNG',
           '1956 cc | 17 kmpl | Diesel',
           '1956 cc | 16 kmpl | Diesel',
           '1199 cc | 19 kmpl | Petrol | CNG',
           '1199 cc | 24 kmpl | Petrol | Diesel | CNG',
           '1199 cc | 20 kmpl | Petrol | CNG',
           '2956 cc | 15 kmpl | Diesel',
           '1199 cc | 20 kmpl | Petrol | CNG']
```

7.Accessing the Ratings:



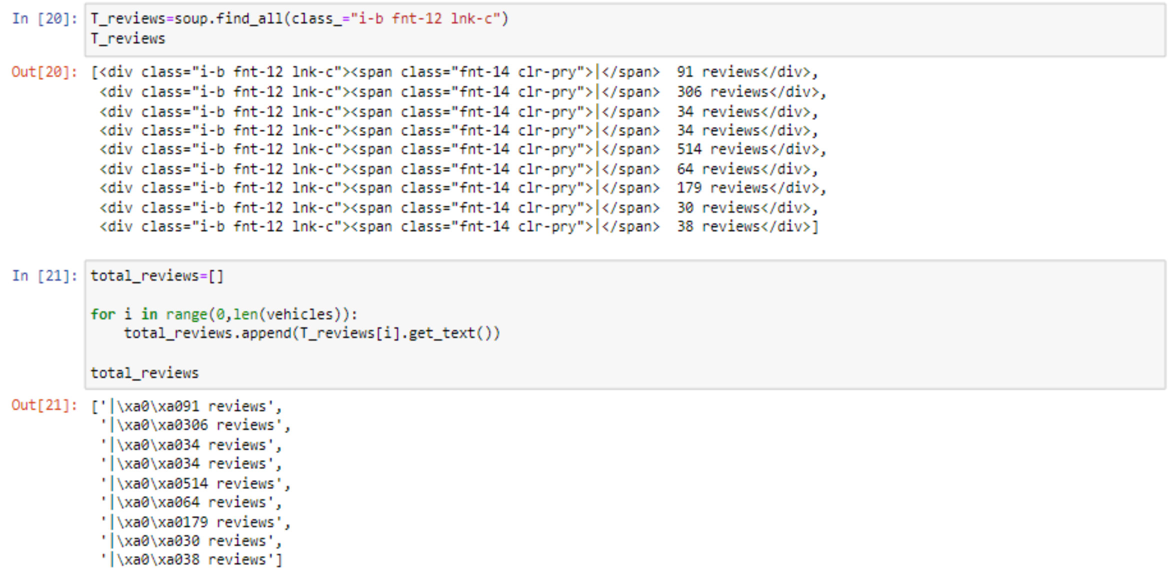
```
In [16]: Ratings=soup.find_all(class_="r-w fnt-12 rel i-b rt-g")
         Rating=soup.find_all(class_="r-w fnt-12 rel i-b rt-lg")
         print(Ratings)
         Rating

         [<div class="r-w fnt-12 rel i-b rt-g">4.5</div>, <div class="r-w fnt-12 rel i-b rt-g">4.4</div>, <div class="r-w fnt-12 rel i-b
         rt-g">4.6</div>, <div class="r-w fnt-12 rel i-b rt-g">4.8</div>, <div class="r-w fnt-12 rel i-b rt-g">4.3</div>, <div class="r-
         w fnt-12 rel i-b rt-g">4.1</div>, <div class="r-w fnt-12 rel i-b rt-g">4.3</div>, <div class="r-w fnt-12 rel i-b rt-g">4.3</div
         >]

Out[16]: [<div class="r-w fnt-12 rel i-b rt-lg">3.9</div>]
```

8.Inserting that one rating which is in the different class with the help of enumerate() & insert() function:



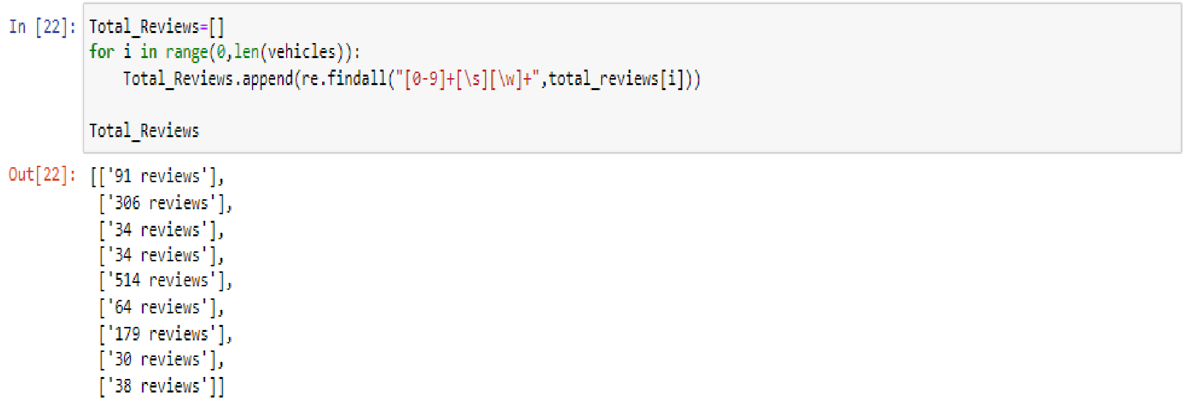
```
In [17]: # Specify the indices where you want to insert elements from Rating
         index_to_insert = [7]

         # Use a loop to insert elements from Rating into Ratings at the specified indices
         for i, item in enumerate(Rating):
             Ratings.insert(index_to_insert[i], item)

         # Print the updated Ratings
         print(Ratings)

         [<div class="r-w fnt-12 rel i-b rt-g">4.5</div>, <div class="r-w fnt-12 rel i-b rt-g">4.4</div>, <div class="r-w fnt-12 rel i-b
         rt-g">4.6</div>, <div class="r-w fnt-12 rel i-b rt-g">4.8</div>, <div class="r-w fnt-12 rel i-b rt-g">4.3</div>, <div class="r-
         w fnt-12 rel i-b rt-g">4.1</div>, <div class="r-w fnt-12 rel i-b rt-g">4.3</div>, <div class="r-w fnt-12 rel i-b rt-lg">3.9</di
         v>, <div class="r-w fnt-12 rel i-b rt-g">4.3</div>]

In [18]: len(Ratings)

Out[18]: 9

In [19]: final_ratings=[]
         for i in range(0,len(vehicles)):
             final_ratings.append(Ratings[i].get_text())
         final_ratings

Out[19]: ['4.5', '4.4', '4.6', '4.8', '4.3', '4.1', '4.3', '3.9', '4.3']
```

9.Accessing the Total Number Of Reviews:



```
In [20]: T_reviews=soup.find_all(class_="i-b fnt-12 lnk-c")
         T_reviews

Out[20]: [<div class="i-b fnt-12 lnk-c"><span class="fnt-14 clr-pry">|</span>  91 reviews</div>,
          <div class="i-b fnt-12 lnk-c"><span class="fnt-14 clr-pry">|</span>  306 reviews</div>,
          <div class="i-b fnt-12 lnk-c"><span class="fnt-14 clr-pry">|</span>  34 reviews</div>,
          <div class="i-b fnt-12 lnk-c"><span class="fnt-14 clr-pry">|</span>  34 reviews</div>,
          <div class="i-b fnt-12 lnk-c"><span class="fnt-14 clr-pry">|</span>  514 reviews</div>,
          <div class="i-b fnt-12 lnk-c"><span class="fnt-14 clr-pry">|</span>  64 reviews</div>,
          <div class="i-b fnt-12 lnk-c"><span class="fnt-14 clr-pry">|</span>  179 reviews</div>,
          <div class="i-b fnt-12 lnk-c"><span class="fnt-14 clr-pry">|</span>  30 reviews</div>,
          <div class="i-b fnt-12 lnk-c"><span class="fnt-14 clr-pry">|</span>  38 reviews</div>]

In [21]: total_reviews=[]

         for i in range(0,len(vehicles)):
             total_reviews.append(T_reviews[i].get_text())

         total_reviews

Out[21]: ['|\xa0\xa091 reviews',
          '|\xa0\xa0306 reviews',
          '|\xa0\xa034 reviews',
          '|\xa0\xa034 reviews',
          '|\xa0\xa0514 reviews',
          '|\xa0\xa064 reviews',
          '|\xa0\xa0179 reviews',
          '|\xa0\xa030 reviews',
          '|\xa0\xa038 reviews']
```

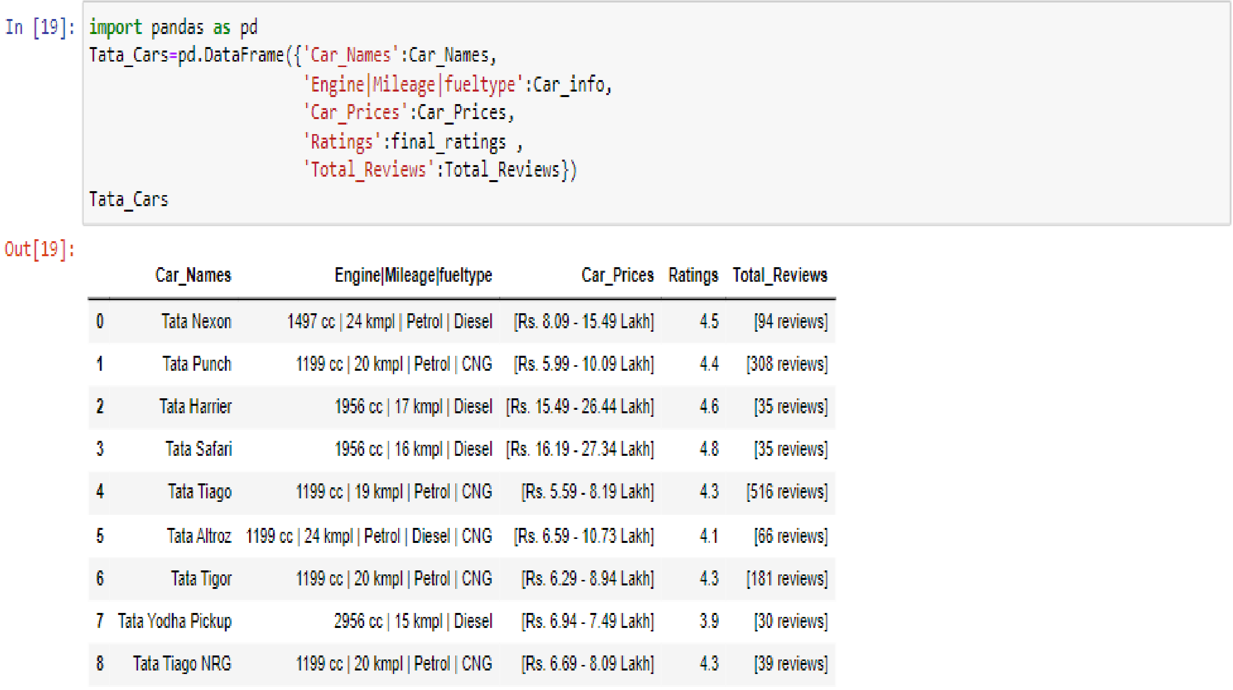
10. Using RegEx to get the required text(Total_Reviews) from the list:



In [22]:
```
Total_Reviews=[]
for i in range(0,len(vehicles)):
    Total_Reviews.append(re.findall("[0-9]+[\s][\w]+",total_reviews[i]))

Total_Reviews
```

Out[22]:
```
[['91 reviews'],
 ['306 reviews'],
 ['34 reviews'],
 ['34 reviews'],
 ['514 reviews'],
 ['64 reviews'],
 ['179 reviews'],
 ['30 reviews'],
 ['38 reviews']]
```

11.Importing the pandas and creating DataFrame:

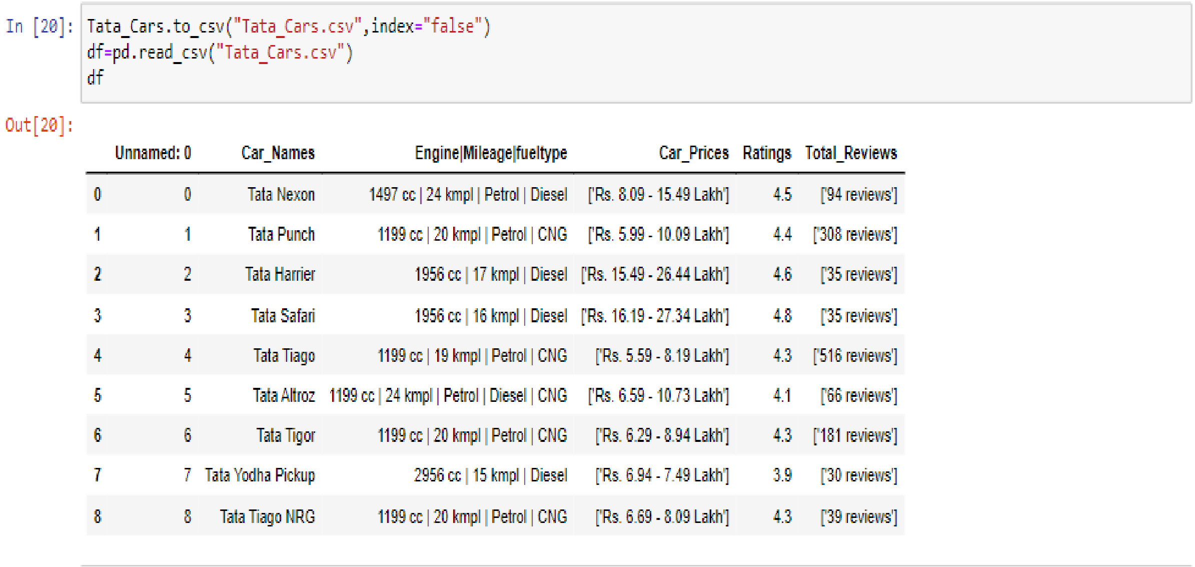


In [19]:
```
import pandas as pd
Tata_Cars=pd.DataFrame({'Car_Names':Car_Names,
                        'Engine|Mileage|fueltype':Car_info,
                        'Car_Prices':Car_Prices,
                        'Ratings':final_ratings ,
                        'Total_Reviews':Total_Reviews})
Tata_Cars
```

Out[19]:

	Car_Names	Engine|Mileage|fueltype	Car_Prices	Ratings	Total_Reviews
0	Tata Nexon	1497 cc | 24 kmpl | Petrol | Diesel	[Rs. 8.09 - 15.49 Lakh]	4.5	[94 reviews]
1	Tata Punch	1199 cc | 20 kmpl | Petrol | CNG	[Rs. 5.99 - 10.09 Lakh]	4.4	[308 reviews]
2	Tata Harrier	1956 cc | 17 kmpl | Diesel	[Rs. 15.49 - 26.44 Lakh]	4.6	[35 reviews]
3	Tata Safari	1956 cc | 16 kmpl | Diesel	[Rs. 16.19 - 27.34 Lakh]	4.8	[35 reviews]
4	Tata Tiago	1199 cc | 19 kmpl | Petrol | CNG	[Rs. 5.59 - 8.19 Lakh]	4.3	[516 reviews]
5	Tata Altroz	1199 cc | 24 kmpl | Petrol | Diesel | CNG	[Rs. 6.59 - 10.73 Lakh]	4.1	[66 reviews]
6	Tata Tigor	1199 cc | 20 kmpl | Petrol | CNG	[Rs. 6.29 - 8.94 Lakh]	4.3	[181 reviews]
7	Tata Yodha Pickup	2956 cc | 15 kmpl | Diesel	[Rs. 6.94 - 7.49 Lakh]	3.9	[30 reviews]
8	Tata Tiago NRG	1199 cc | 20 kmpl | Petrol | CNG	[Rs. 6.69 - 8.09 Lakh]	4.3	[39 reviews]

12.converting and Storing DataFrame in the form of a CSV file and opening the file in application:

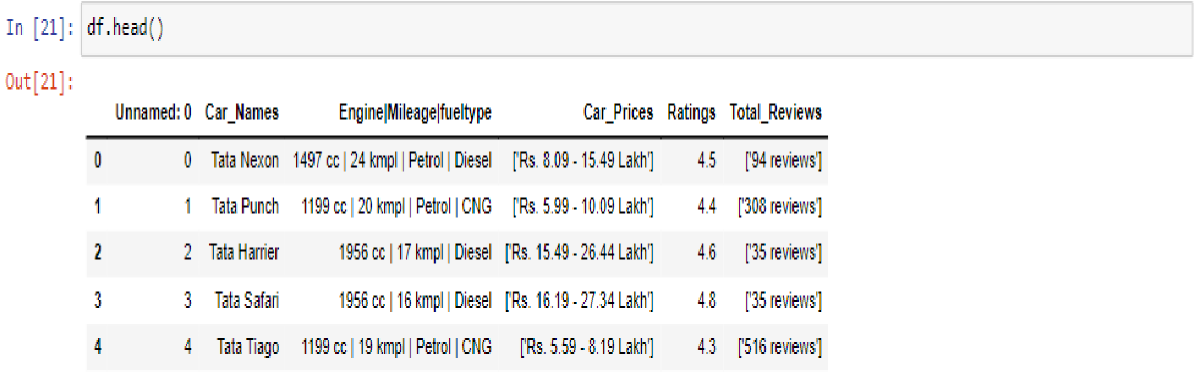


In [20]:
```
Tata_Cars.to_csv("Tata_Cars.csv",index="false")
df=pd.read_csv("Tata_Cars.csv")
df
```

Out[20]:

	Unnamed: 0	Car_Names	Engine|Mileage|fueltype	Car_Prices	Ratings	Total_Reviews
0	0	Tata Nexon	1497 cc | 24 kmpl | Petrol | Diesel	['Rs. 8.09 - 15.49 Lakh']	4.5	['94 reviews']
1	1	Tata Punch	1199 cc | 20 kmpl | Petrol | CNG	['Rs. 5.99 - 10.09 Lakh']	4.4	['308 reviews']
2	2	Tata Harrier	1956 cc | 17 kmpl | Diesel	['Rs. 15.49 - 26.44 Lakh']	4.6	['35 reviews']
3	3	Tata Safari	1956 cc | 16 kmpl | Diesel	['Rs. 16.19 - 27.34 Lakh']	4.8	['35 reviews']
4	4	Tata Tiago	1199 cc | 19 kmpl | Petrol | CNG	['Rs. 5.59 - 8.19 Lakh']	4.3	['516 reviews']
5	5	Tata Altroz	1199 cc | 24 kmpl | Petrol | Diesel | CNG	['Rs. 6.59 - 10.73 Lakh']	4.1	['66 reviews']
6	6	Tata Tigor	1199 cc | 20 kmpl | Petrol | CNG	['Rs. 6.29 - 8.94 Lakh']	4.3	['181 reviews']
7	7	Tata Yodha Pickup	2956 cc | 15 kmpl | Diesel	['Rs. 6.94 - 7.49 Lakh']	3.9	['30 reviews']
8	8	Tata Tiago NRG	1199 cc | 20 kmpl | Petrol | CNG	['Rs. 6.69 - 8.09 Lakh']	4.3	['39 reviews']

13.Accessing first five rows of DataFrame Using head():

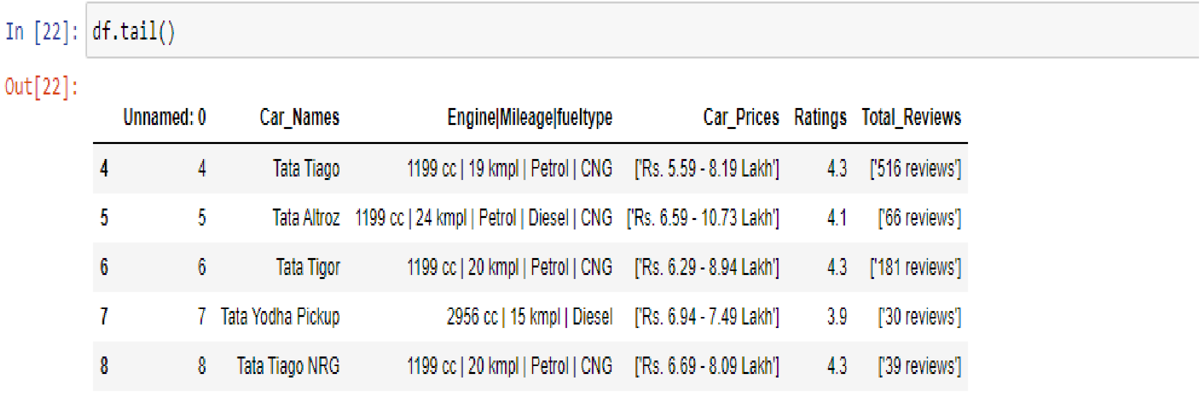


In [21]:
```
df.head()
```

Out[21]:

	Unnamed: 0	Car_Names	Engine|Mileage|fueltype	Car_Prices	Ratings	Total_Reviews
0	0	Tata Nexon	1497 cc | 24 kmpl | Petrol | Diesel	['Rs. 8.09 - 15.49 Lakh']	4.5	['94 reviews']
1	1	Tata Punch	1199 cc | 20 kmpl | Petrol | CNG	['Rs. 5.99 - 10.09 Lakh']	4.4	['308 reviews']
2	2	Tata Harrier	1956 cc | 17 kmpl | Diesel	['Rs. 15.49 - 26.44 Lakh']	4.6	['35 reviews']
3	3	Tata Safari	1956 cc | 16 kmpl | Diesel	['Rs. 16.19 - 27.34 Lakh']	4.8	['35 reviews']
4	4	Tata Tiago	1199 cc | 19 kmpl | Petrol | CNG	['Rs. 5.59 - 8.19 Lakh']	4.3	['516 reviews']

14.Accessing Last five rows of DataFrame Using tail():

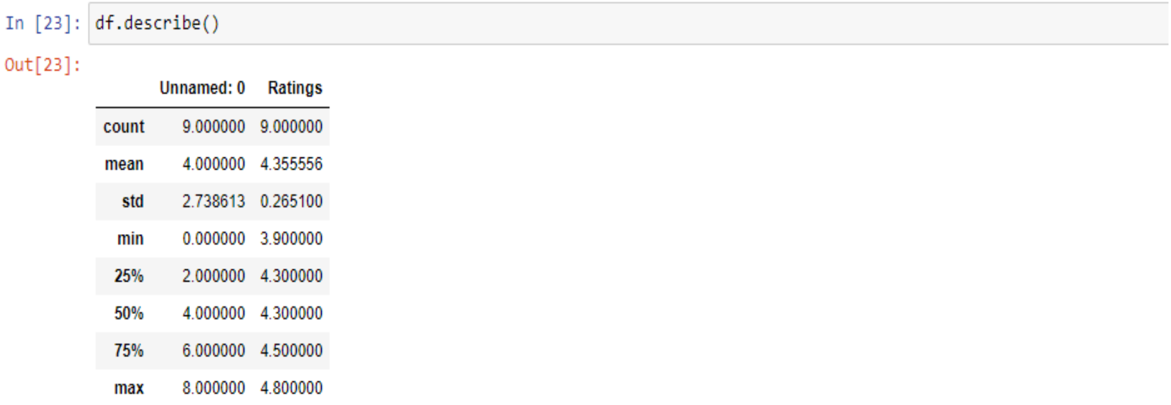


In [22]:
```
df.tail()
```

Out[22]:

	Unnamed: 0	Car_Names	Engine|Mileage|fueltype	Car_Prices	Ratings	Total_Reviews
4	4	Tata Tiago	1199 cc | 19 kmpl | Petrol | CNG	['Rs. 5.59 - 8.19 Lakh']	4.3	['516 reviews']
5	5	Tata Altroz	1199 cc | 24 kmpl | Petrol | Diesel | CNG	['Rs. 6.59 - 10.73 Lakh']	4.1	['66 reviews']
6	6	Tata Tigor	1199 cc | 20 kmpl | Petrol | CNG	['Rs. 6.29 - 8.94 Lakh']	4.3	['181 reviews']
7	7	Tata Yodha Pickup	2956 cc | 15 kmpl | Diesel	['Rs. 6.94 - 7.49 Lakh']	3.9	['30 reviews']
8	8	Tata Tiago NRG	1199 cc | 20 kmpl | Petrol | CNG	['Rs. 6.69 - 8.09 Lakh']	4.3	['39 reviews']

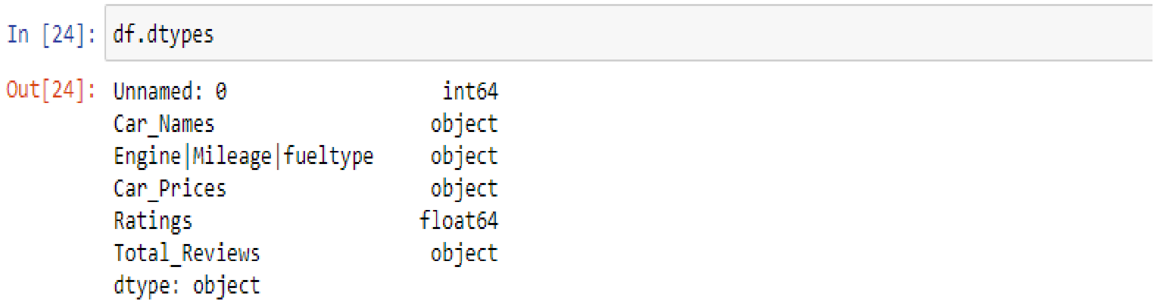
15.Using the describe() function to get information about numerical columns:



```
In [23]: df.describe()
Out[23]:
```

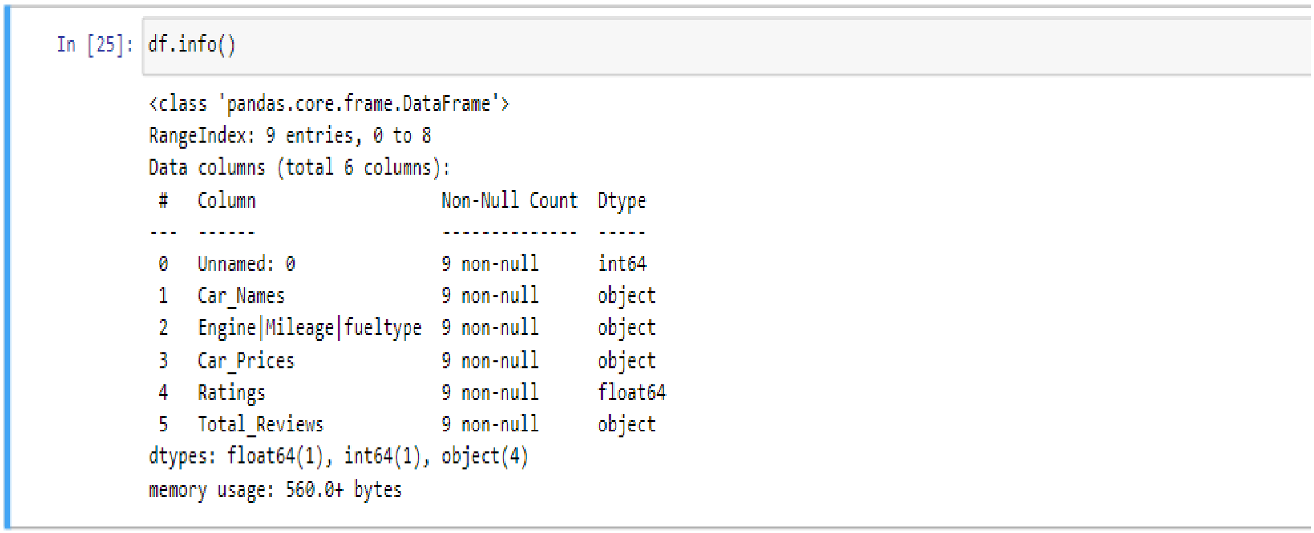
	Unnamed: 0	Ratings
count	9.000000	9.000000
mean	4.000000	4.355556
std	2.738613	0.265100
min	0.000000	3.900000
25%	2.000000	4.300000
50%	4.000000	4.300000
75%	6.000000	4.500000
max	8.000000	4.800000

16.Using dtypes to get the data type of each column:



```
In [24]: df.dtypes

Out[24]: Unnamed: 0                   int64
         Car_Names                   object
         Engine|Mileage|fueltype     object
         Car_Prices                  object
         Ratings                    float64
         Total_Reviews               object
         dtype: object
```

17.Using info() to get information about the DataFrame:



```
In [25]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 6 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Unnamed: 0               9 non-null      int64
 1   Car_Names                9 non-null      object
 2   Engine|Mileage|fueltype  9 non-null      object
 3   Car_Prices               9 non-null      object
 4   Ratings                  9 non-null      float64
 5   Total_Reviews            9 non-null      object
dtypes: float64(1), int64(1), object(4)
memory usage: 560.0+ bytes
```

6.Conclusion

In conclusion, the web scraping project focused on Tata Motors cars from ZigWheels.com has successfully achieved its objectives. The comprehensive data retrieval provided detailed specifications, pricing information, and user sentiments, enriching our understanding of Tata Motors' offerings. Ethical compliance and data quality assurance were diligently maintained throughout the scraping process, ensuring the reliability and integrity of the dataset. The comparative analysis with competitors unveiled Tata Motors' standing in the market. Trends identified in user preferences contribute valuable insights for anticipating market shifts. Clear documentation of the scraping methodology and codebase enhances the project's transparency and future reference. This project forms a solid foundation for strategic decision-making and further exploration within the automotive industry.