Car Price Prediction Project - Part 1 Web Scraping Data of Used Car

Strategy :

1. Selenium will be used for webscraping data from cardheko.com
2. In first part Scraping URL of Used car for different location in India.
3. Storing Scrap URL in excel file.
4. Selecting car feature to be scrap from website.
5. In second part Scraping data from indiviual URL in excel file.
6. Exporting final data in Excel file.

Part 1 : Scraping URLs of Used car from Cardheko.com

- **Importing libraries require for scraping**

In [1]:
```
import pandas as pd
import numpy as pd
import time
import selenium
from selenium import webdriver
from selenium.common.exceptions import StaleElementReferenceException, NoSuchE
```

- **Importing webdriver**

In [2]:
```
driver=webdriver.Chrome(r'C:\chromedriver.exe')
```

- **Opening cardheko website in browser**

In [113]:
```
url = "https://www.cardekho.com/"
driver.get(url)
time.sleep(2)
```

In [16]:
```
Used_cars=driver.find_element_by_xpath('//li[@data-slug="/usedCars"]/a').get_a
driver.get(Used_cars)
time.sleep(2)
```

- **Collecting url of different location/ city for futher scraping**

1. Extracting data for Ahmedabad city

In [16]:
```
url = "https://www.cardekho.com/used-cars+in+ahmedabad"
driver.get(url)
time.sleep(2)
```

In [17]:
```
from tqdm import tqdm
for _ in tqdm(range(0,200)):
    time.sleep(0.5)
    driver.execute_script("window.scrollBy(0,1000)","")
    time.sleep(1)
    driver.execute_script("window.scrollBy(0,-350)")
```

```
100%|██████████| 200/200 [05:42<00:00,  1.71s/it]
```

In [18]:
```
Car_url_ahmedabad = []
car_url_ahmedabad = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-7
for j in tqdm(range(len(car_url_ahmedabad))):
    Car_url_ahmedabad.append(car_url_ahmedabad[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 509/509 [00:04<00:00, 112.64it/s]
```

In [20]:
```
len(Car_url_ahmedabad)
```

Out[20]: 509

2. Extracting URL for Bangalore city

In [33]:
```
url = "https://www.cardekho.com/used-cars+in+bangalore"
driver.get(url)
time.sleep(2)
```

In [34]:
```
from tqdm import tqdm
for _ in tqdm(range(0,300)):
    time.sleep(0.75)
    driver.execute_script("window.scrollBy(0,1000)","")
    time.sleep(1)
    driver.execute_script("window.scrollBy(0,-350)")
```

```
100%|██████████| 300/300 [09:45<00:00,  1.95s/it]
```

In [35]:
```
Car_url_bangalore = []
car_url_bangalore = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-7
for j in tqdm(range(len(car_url_bangalore))):
    Car_url_bangalore.append(car_url_bangalore[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 580/580 [00:05<00:00, 105.58it/s]
```

In [36]:
```
len(Car_url_bangalore)
```

Out[36]: 580

3. Extracting URL for Chennai

In [37]:
```
url = "https://www.cardekho.com/used-cars+in+chennai"
driver.get(url)
time.sleep(2)
```

In [38]:
```
from tqdm import tqdm
for _ in tqdm(range(0,250)):
    time.sleep(0.5)
    driver.execute_script("window.scrollBy(0,1000)","")
    time.sleep(1)
    driver.execute_script("window.scrollBy(0,-350)")
```

```
100%|██████████| 250/250 [06:35<00:00,  1.58s/it]
```

In [39]:
```
Car_url_chennai = []
car_url_chennai = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-7  c
for j in tqdm(range(len(car_url_chennai))):
    Car_url_chennai.append(car_url_chennai[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 298/298 [00:02<00:00, 117.45it/s]
```

4. Extracting URL for Dehli-NCR

In [48]:
```
url = "https://www.cardekho.com/used-cars+in+delhi-ncr"
driver.get(url)
time.sleep(2)
```

In [49]:
```
from tqdm import tqdm
for _ in tqdm(range(0,1100)):
    time.sleep(0.5)
    driver.execute_script("window.scrollBy(0,1500)","")
    time.sleep(1)
    driver.execute_script("window.scrollBy(0,-500)")
```

```
100%|██████████| 1100/1100 [53:55<00:00,  2.94s/it]
```

In [50]:
```
Car_url_delhi_ncr = []
car_url_delhi_ncr = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-7
for j in tqdm(range(len(car_url_delhi_ncr))):
    Car_url_delhi_ncr.append(car_url_delhi_ncr[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 3141/3141 [01:03<00:00, 49.52it/s]
```

5. Extracting URL for Gurgaon city

In [51]:
```
url = "https://www.cardekho.com/used-cars+in+gurgaon"
driver.get(url)
time.sleep(2)
```

In [52]:
```
from tqdm import tqdm
for _ in tqdm(range(0,600)):
    time.sleep(0.5)
    driver.execute_script("window.scrollBy(0,1500)","")
    time.sleep(1)
    driver.execute_script("window.scrollBy(0,-500)")
```

```
100%|██████████| 600/600 [20:22<00:00,  2.04s/it]
```

In [53]:
```
Car_url_gurgaon = []
car_url_gurgaon = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-7  c
for j in tqdm(range(len(car_url_gurgaon))):
    Car_url_gurgaon.append(car_url_gurgaon[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 1217/1217 [00:12<00:00, 100.96it/s]
```

6. Extracting URL for Telangana

In [54]:
```
url = "https://www.cardekho.com/used-cars+in+telangana"
driver.get(url)
time.sleep(2)
```

In [56]:
```
from tqdm import tqdm
for _ in tqdm(range(0,400)):
    time.sleep(0.5)
    driver.execute_script("window.scrollBy(0,1500)","")
    time.sleep(1)
    driver.execute_script("window.scrollBy(0,-500)")
```

```
100%|██████████| 400/400 [17:11<00:00,  2.58s/it]
```

In [57]:
```
Car_url_telangana = []
car_url_telangana = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-7
for j in tqdm(range(len(car_url_telangana))):
    Car_url_telangana.append(car_url_telangana[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 1210/1210 [00:10<00:00, 110.06it/s]
```

7. Extracting URL for Maharashtra

In [61]:
```
url = "https://www.cardekho.com/used-cars+in+maharashtra"
driver.get(url)
time.sleep(2)
```

In [62]:
```
from tqdm import tqdm
for _ in tqdm(range(0,1000)):
    driver.execute_script("window.scrollBy(0,1500)","")
    time.sleep(0.5)
    driver.execute_script("window.scrollBy(0,-500)")
```

```
100%|██████████| 1000/1000 [09:30<00:00,  1.75it/s]
```

In [60]:
```
Car_url_Maharashtra = []
car_url_Maharashtra = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-
for j in tqdm(range(len(car_url_Maharashtra))):
    Car_url_Maharashtra.append(car_url_Maharashtra[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 4526/4526 [01:24<00:00, 53.65it/s]
```

8. Extracting URL for Karnataka

In [63]:
```
url = "https://www.cardekho.com/used-cars+in+karnataka"
driver.get(url)
time.sleep(2)
```

In [64]:
```
from tqdm import tqdm
for _ in tqdm(range(0,750)):

    driver.execute_script("window.scrollBy(0,1500)","")
    time.sleep(0.4)
    driver.execute_script("window.scrollBy(0,-500)")
```

```
100%|██████████| 750/750 [08:08<00:00,  1.53it/s]
```

In [65]:
```
Car_url_Karnataka = []
car_url_Karnataka = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-7
for j in tqdm(range(len(car_url_Karnataka))):
    Car_url_Karnataka.append(car_url_Karnataka[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 867/867 [00:07<00:00, 111.79it/s]
```

9. Extracting URL for Uttar Pradesh

In [66]:
```
url = "https://www.cardekho.com/used-cars+in+uttar-pradesh"
driver.get(url)
time.sleep(2)
```

In [67]:
```
from tqdm import tqdm
for _ in tqdm(range(0,700)):
    driver.execute_script("window.scrollBy(0,1500)","")
    time.sleep(0.25)
    driver.execute_script("window.scrollBy(0,-500)")
```

```
100%|██████████| 700/700 [10:19<00:00,  1.13it/s]
```

In [68]:
```
Car_url_UttarPradesh = []
car_url_UttarPradesh = driver.find_elements_by_xpath('//div[@class="gsc_col-xs
for j in tqdm(range(len(car_url_UttarPradesh))):
    Car_url_UttarPradesh.append(car_url_UttarPradesh[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 1380/1380 [00:15<00:00, 89.31it/s]
```

10. Extracting URL for Tamil Nadu

In [69]:
```
url = "https://www.cardekho.com/used-cars+in+tamil-nadu"
driver.get(url)
time.sleep(2)
```

In [ ]:
```
from tqdm import tqdm
for _ in tqdm(range(0,600)):
    driver.execute_script("window.scrollBy(0,1500)","")
    time.sleep(0.25)
    driver.execute_script("window.scrollBy(0,-500)")
```

In [73]:
```
Car_url_TamilNadu = []
car_url_TamilNadu = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-7
for j in tqdm(range(len(car_url_TamilNadu))):
    Car_url_TamilNadu.append(car_url_TamilNadu[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 1750/1750 [02:35<00:00, 11.24it/s]
```

11. Extracting URL for Haryana

In [75]:
```
url = "https://www.cardekho.com/used-cars+in+haryana"
driver.get(url)
time.sleep(2)
```

In [ ]:
```
from tqdm import tqdm
for _ in tqdm(range(0,600)):

    driver.execute_script("window.scrollBy(0,1500)","")
    time.sleep(0.25)
    driver.execute_script("window.scrollBy(0,-500)")
```

In [79]:
```
Car_url_Haryana = []
car_url_Haryana = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-7  c
for j in tqdm(range(len(car_url_Haryana))):
    Car_url_Haryana.append(car_url_Haryana[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 1228/1228 [00:12<00:00, 96.30it/s]
```

In [99]:
```
len(Car_url_Haryana)
```

Out[99]:
```
1228
```

12. Extracting URL for Rajasthan

In [80]:
```
url = "https://www.cardekho.com/used-cars+in+rajasthan"
driver.get(url)
time.sleep(2)
```

In [81]:
```
from tqdm import tqdm
for _ in tqdm(range(0,500)):

    driver.execute_script("window.scrollBy(0,1500)","")
    time.sleep(0.25)
    driver.execute_script("window.scrollBy(0,-500)")
```

```
100%|██████████| 500/500 [04:49<00:00,  1.73it/s]
```

In [82]:
```
Car_url_Rajasthan = []
car_url_Rajasthan = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-7
for j in tqdm(range(len(car_url_Rajasthan))):
    Car_url_Rajasthan.append(car_url_Rajasthan[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 687/687 [00:06<00:00, 111.65it/s]
```

13. Extracting URL for Kerala

In [83]:
```
url = "https://www.cardekho.com/used-cars+in+kerala"
driver.get(url)
time.sleep(2)
```

In [84]:
```
from tqdm import tqdm
for _ in tqdm(range(0,400)):
    time.sleep(0.5)
    driver.execute_script("window.scrollBy(0,1500)","")
    time.sleep(1)
    driver.execute_script("window.scrollBy(0,-500)")
```

```
100%|██████████| 400/400 [10:11<00:00,  1.53s/it]
```

In [85]:
```
Car_url_Kerala = []
car_url_Kerala = driver.find_elements_by_xpath('//div[@class="gsc_col-xs-7  ca
for j in tqdm(range(len(car_url_Kerala))):
    Car_url_Kerala.append(car_url_Kerala[j].get_attribute('href'))
time.sleep(2)
```

```
100%|██████████| 18/18 [00:00<00:00, 117.58it/s]
```

In [87]:
```
Car_url = []
```

In [89]:
```
Car_url = Car_url_Kerala + Car_url_Rajasthan + Car_url_Haryana + Car_url_Tamil
```

In [90]:
```
len(Car_url)
```

Out[90]:
```
12305
```

Creating Excel file of URL for futher webscraping

In [3]:
```
import pandas as pd
```

In [97]:
```
Car_URL = pd.DataFrame({})
Car_URL['Urls'] = Car_url
```

In [98]:
```
Car_URL.to_excel('Car_url.xlsx', index = False)
```

Part 2 : Scraping features from Indiviudal Link

Importing excel file contain URLs.

In [6]:
```
import pandas as pd
```

In [4]:
```
df = pd.read_excel('Car_url.xlsx')
```

In [5]:
```
df.shape
```

Out[5]:
```
(12305, 1)
```

As we have scrap around 12035 URL, We Will scrap car details in different batchs.

In [6]:
```
# Making Empty lists
Location = []
Model = []
Variant = []
Price = []
Make_year =[]
Fuel_Type = []
KMs_driven = []
Engine_displacement = []
Transmission = []
Milage = []
Max_power = []
Torque = []
Seats = []
Color = []
Gear_Box =[]
Steering_Type =[]
Front_Brake_Type = []
Rear_Brake_Type = []
Tyre_Volume = []
Cargo_volume = []
Engine_Type = []
No_of_cylinder = []
Value_Configuration = []
Fuel_Suppy_System = []
Turbo_charger = []
Super_charger = []
Length = []
Width =[]
Height = []
Gross_weight = []
```

In [21]:
```
driver=webdriver.Chrome(r'C:\chromedriver.exe')
```

Extracting details for batch 1 of 500

In [16]:
```
from tqdm import tqdm
for i in tqdm(df['Urls'][9800:9900]):
    driver.get(i)
    time.sleep(0.5)

    # Extracting Car Model via xpath
    try :
        model = driver.find_element_by_xpath('//div[@class="gsc_col-xs-12"]/h1
        Model.append(model.text[5:])
    except NoSuchElementException:
        try :
            model = driver.find_element_by_xpath('//div[@class="gsc_container_
            Model.append(model.text)
        except NoSuchElementException:
            pass

    #clicking to view all specifications
    try:
        view_more = driver.find_element_by_xpath("//*[text() = 'View All Speci
        driver.execute_script("arguments[0].scrollIntoView();", view_more)
        driver.execute_script("arguments[0].click();", view_more)

    except NoSuchElementException:
        try:
            Button= driver.find_element_by_xpath('//*[@id="topspec"]/div[2]/a'
            Button.click()
            time.sleep(1)
        except NoSuchElementException:
            pass

    time.sleep(0.75)
    # Extracting Car Price via xpath
    try :
        price = driver.find_element_by_xpath('//div[@class="priceSection"]/spa
        Price.append(price.text)
    except NoSuchElementException:
        try :
            price = driver.find_element_by_xpath('//div[@class="gsc_container_
            Price.append(price.text)
        except NoSuchElementException:
            pass

    # Extracting Car Make Year via xpath
    try :
        year = driver.find_element_by_xpath('//*[text()="Make Year"]/following
        Make_year.append(year.text)
    except NoSuchElementException:
        try :
            year = driver.find_element_by_xpath('//div[@class="GenDetailBox"]/
            Make_year.append(year.text)
        except NoSuchElementException:
            pass

    # Extracting Car Fuel Type via xpath
    try :
        fuel = driver.find_element_by_xpath('//*[text()="Fuel"]/following-sibl
        Fuel_Type.append(fuel.text)
```

```
    except NoSuchElementException:
        try :
            fuel = driver.find_element_by_xpath('//div[@class="GenDetailBox"]/
            Fuel_Type.append(fuel.text)
        except NoSuchElementException:
            Fuel_Type.append('-')

    # Extracting KMS driven via xpath
    try :
        kms = driver.find_element_by_xpath('//*[text()="KMs Driven"]/following
        KMs_driven.append(kms.text.replace('Kms',''))
    except NoSuchElementException:
        try :
            kms = driver.find_element_by_xpath('//div[@class="GenDetailBox"]/u
            KMs_driven.append(kms.text.replace('kms',''))
        except NoSuchElementException:
            pass

    # Extracting Engine_displacemet via xpath
    try :
        engine_disp = driver.find_element_by_xpath('//*[text()="Engine Displac
        Engine_displacement.append(engine_disp.text.replace('CC',''))
    except NoSuchElementException:
        try :
            engine_disp = driver.find_element_by_xpath('//*[text()="Engine"]/f
            Engine_displacement.append(engine_disp.text.replace('CC',''))
        except NoSuchElementException:
            pass

    # Extracting Transmission via xpath
    try :
        transmission = driver.find_element_by_xpath('//*[text()="Transmission"
        Transmission.append(transmission.text)
    except NoSuchElementException:
        try :
            transmission = driver.find_element_by_xpath('//div[@class="GenDeta
            Transmission.append(transmission.text)
        except NoSuchElementException:
            pass
    time.sleep(0.25)
    # Extracting Milage via xpath
    try :
        milage = driver.find_element_by_xpath('//*[text()="Mileage"]/following
        Milage.append(milage.text.replace('kmpl',''))

    except NoSuchElementException:
        Milage.append('-')

    # Extracting Max_power via xpath
    try :
        maxbhp = driver.find_element_by_xpath('//*[text()="Max Power"]/followi
        Max_power.append(maxbhp.text.replace('bhp',''))

    except NoSuchElementException:
        Max_power.append('-')

   # Extracting Torque via xpath
```

```
    try :
        torque = driver.find_element_by_xpath('//*[text()="Torque"]/following-
        Torque.append(torque.text.replace('Nm',''))

    except NoSuchElementException:
        Torque.append('-')

    # Extracting Seating capacity via xpath
    try :
        seats = driver.find_element_by_xpath('//*[text()="Seating Capacity"]/f
        Seats.append(seats.text)

    except NoSuchElementException:
        Seats.append('-')

    # Extracting color via xpath
    try :
        color = driver.find_element_by_xpath('//*[text()="Color"]/following-si
        Color.append(color.text)
    except NoSuchElementException:
        Color.append('-')

    # Extracting Gear_Box via xpath
    try :
        gear_Box = driver.find_element_by_xpath('//*[text()="Gear Box"]/follow
        Gear_Box.append(gear_Box.text)

    except NoSuchElementException:
        Gear_Box.append('-')

    # Extracting Steering_Type via xpath
    try :
        steering_Type = driver.find_element_by_xpath('//*[text()="Steering Typ
        Steering_Type.append(steering_Type.text)

    except NoSuchElementException:
        Steering_Type.append('-')

    # Extracting Front_Brake_Type via xpath
    try :
        front_Brake_Type = driver.find_element_by_xpath('//*[text()="Front Bra
        Front_Brake_Type.append(front_Brake_Type.text)

    except NoSuchElementException:
        Front_Brake_Type.append('-')

    # Extracting Rear_Brake_Type via xpath
    try :
        rear_Brake_Type = driver.find_element_by_xpath('//*[text()="Rear Brake
        Rear_Brake_Type.append(rear_Brake_Type.text)

    except NoSuchElementException:
        Rear_Brake_Type.append('-')

   # Extracting Tyre_Volume via xpath
    try :
        tyre_Volume = driver.find_element_by_xpath('//*[text()="Tyre Type"]/fo
```

```
        Tyre_Volume.append(tyre_Volume.text)

    except NoSuchElementException:
        Tyre_Volume.append('-')

    # Extracting Engine_Type via xpath
    try :
        engine_Type = driver.find_element_by_xpath('//*[text()="Engine Type"]/
        Engine_Type.append(engine_Type.text)

    except NoSuchElementException:
        Engine_Type.append('-')

    # Extracting No_of_cylinder via xpath
    try :
        no_of_cylinder = driver.find_element_by_xpath('//*[text()="No of Cylin
        No_of_cylinder.append(no_of_cylinder.text)
    except NoSuchElementException:
        try :
            no_of_cylinder = driver.find_element_by_xpath('//*[text()="No Of C
            No_of_cylinder.append(no_of_cylinder.text)
        except NoSuchElementException:
            pass

    # Extracting Value_Configuration via xpath
    try :
        value_Configuration = driver.find_element_by_xpath('//*[text()="Value
        Value_Configuration.append(value_Configuration.text)
    except NoSuchElementException:
        try :
            value_Configuration = driver.find_element_by_xpath('//*[text()="Va
            Value_Configuration.append(value_Configuration.text)
        except NoSuchElementException:
            pass

    # Extracting Turbo_charger via xpath
    try :
        turbo_charger = driver.find_element_by_xpath('//*[text()="Turbo Charge
        Turbo_charger.append(turbo_charger.text)

    except NoSuchElementException:
        Turbo_charger.append('-')

    # Extracting Super_charger via xpath
    try :
        super_charger = driver.find_element_by_xpath('//*[text()="Super Charge
        Super_charger.append(super_charger.text)
    except NoSuchElementException:
        try :
            super_charger = driver.find_element_by_xpath('//*[text()="superCha
            Super_charger.append(super_charger.text)
        except NoSuchElementException:
            Super_charger.append('-')

    # Extracting Length via xpath
    try :
        length = driver.find_element_by_xpath('//*[text()="Length"]/following-
```

```
        Length.append(length.text.replace('mm',''))

    except NoSuchElementException:
        Length.append('-')

    # Extracting Width via xpath
    try :
        width = driver.find_element_by_xpath('//*[text()="Width"]/following-si
        Width.append(width.text.replace('mm',''))

    except NoSuchElementException:
        Width.append('-')

    # Extracting Height via xpath
    try :
        height = driver.find_element_by_xpath('//*[text()="Height"]/following-
        Height.append(height.text.replace('mm',''))

    except NoSuchElementException:
        Height.append('-')
```

```
100%|██████████| 100/100 [07:57<00:00,  4.78s/it]
```

In [18]:
```
data = list(zip(Model, Make_year, Fuel_Type, KMs_driven, Engine_displacement,
                Max_power, Torque, Seats, Color, Gear_Box, Steering_Type, Front_
                Tyre_Volume, Engine_Type, No_of_cylinder,
                Turbo_charger, Super_charger, Length, Width, Height, Price))
Batch7 = pd.DataFrame(data, columns=['Car Model', 'Make Year', 'Fuel Type', 'K
                                     'Transmission', 'Milage(kmpl)', 'Max Power(
                                     'Color', 'Gear Box', 'Steering Type', 'Fron
                                      'Tyre Volume', 'Engine Type', 'No of Cylin
                                      'Turbo Charger', 'Super Charger', 'Length(
                                     'Height(mm)', 'Price(Rs)'])

pd.set_option('display.max_columns', None)
Batch7.head(5)
```

Out[18]:

	Car Model	Make Year	Fuel Type	KMs driven	Engine Displacement(CC)	Transmission	Milage(kmpl)	Max Power(bhp)
0	BMW 5 Series 520d	2012	Diesel	56,000	1995	Automatic	18.48	177
1	Audi Q3 35 TDI Quattro Premium Plus	2016	Diesel	97,000	1968	Automatic	15.73	174.33
2	Hyundai Creta SX Opt Diesel AT	2021	Diesel	800	1493	Automatic	18.5	113.42
	Maruti							

In [198]:
```
len(Batch1)
```

Out[198]: 500

For Next Batches same code of batch 1 is Re-Run again & again.

In [14]:
```
len(Batch2)
```

Out[14]: 994

In [22]:
```
len(Batch3)
```

Out[22]: 1457

In [27]:
```
len(Batch4)
```

Out[27]: 3436

In [16]:
```
len(Batch5)
```

Out[16]: 2430

In [24]:
```
len(Batch6)
```

Out[24]: 740

In [19]:
```
len(Batch7)
```

Out[19]: 1612

Exporting Batch wise data in Excel file.

In [199]:
```
# Saving Batch1 data in excel file
Batch1.to_excel('Batch1 (0-500).xlsx', index = False)
```

In [15]:
```
# Saving batch 2 data in excel
Batch2.to_excel('Batch1 (500-1500).xlsx', index = False)
```

In [23]:
```
# Saving batch 3 data in excel
Batch3.to_excel('Batch3 (1500-3000).xlsx', index = False)
```

In [28]:
```
# Saving Batch4 data in excel file 3000-5000
Batch4.to_excel('Batch 4 (3000 - 5000).xlsx', index = False)
```

In [17]:
```
# Saving Batch5 data in excel file 5000 - 7500
Batch5.to_excel('Batch 5 (5000).xlsx', index = False)
```

In [25]:
```
# Saving Batch6 data in excel file 7500 - 8500
Batch6.to_excel('Batch 6 (7500-8250).xlsx', index = False)
```

In [20]:
```
# Saving Batch7 data in excel file 7500 - 8500
Batch7.to_excel('Batch 7 (8250-9800).xlsx', index = False)
```

In [21]:
```
Batch7.shape
```

Out[21]: (1612, 24)

Summary :

- **We have scrape more than 11000 cars details with 24 features.**

In [ ]: