Assignment Prompt:
Retail Store Location Scraper

Store:

Asian Paints

Made by:

Ammar Khaleeque

Khaleeque56@gmail.com

+919536990098

Abstract

Web scraping (or data scraping) is a technique used to collect content and data from the internet. To scrape the data from Asian Paints website, Python is used with Selenium and Pandas. The data scraped include name, address, phone number, timings, latitude and longitude of all the Asian Paints store in India. Data is then analysed and the top stores nearest to the user (after entering users coordinates) are calculated.

Selenium is used to dynamically send pin codes from a CSV file to the location text input and retrieve data of all the stores in that pin code, hence iterating through all the pin codes. Pandas is used to store the date in a Data Frame and save it into a CSV file. Pandas is also used in data analysis.

There were some challenges faced while creating this program like extracting the same data again and problem in extracting data from different pin codes etc, which were then later solved with the appropriate solutions.

Table of Contents

List of Figures

Chapter 1

Introduction

Web scraping (or data scraping) is a technique used to collect content and data from the internet. This data is saved in a local file so that it can be manipulated and analysed as needed

In this assignment, web scraping technique is used to gather the data from a dynamic website: Asian Paints. In the given data the following aspects has been covered:

- Store name
- Store address
- Store timings
- Store Coordinates (Latitude/Longitude)
- Store timings

The data is extracted of almost all the stores in India and saved into a CSV file. Then using data analysis techniques, data in analysed and the top stores nearest to the user (after entering users coordinates) are calculated.

1.1.Approach

In this web scraping assignment Python is used, with Pandas and Selenium as libraries. Selenium is used for web automation and Pandas is used for data analysis. With the help of Selenium, almost all the stores of Asian Paints in India are retrieved by sending pin codes of all the district in India from a csv file as input in the location text box of the store locator page of Asian Paints website. Then, the elements of HTML containing the data of stores (name, address, phone number etc) is retrieved and the text or data is extracted from it.

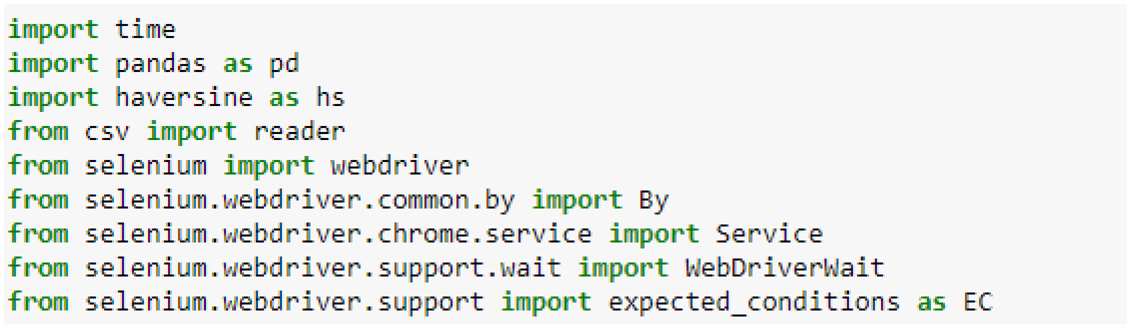
Pandas is used to save the data as a data frame, remove the duplicate stores and then save the data in the csv format. It is also used for data analysis.

Chapter 2

Working Of the Code

In this assignment, Python is used to scrape the data from the Asian Paints website.

2.1. Libraries



```
import time
import pandas as pd
import haversine as hs
from csv import reader
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
```

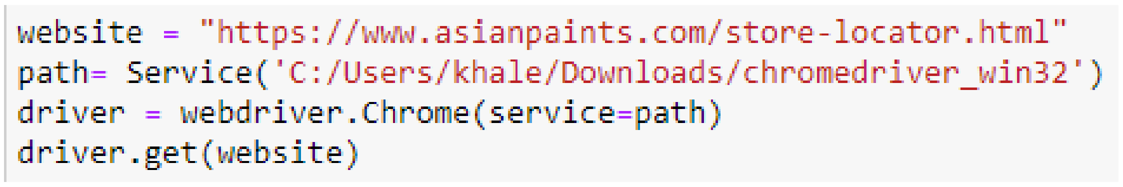
Fig: 2.1. Importing libraries

Multiple libraries are imported to scrape the data from the website. They are as follows:

- *time* is imported to use the method *sleep()*.
- *pandas* is imported to use Data Frames.
- *reader* is imported from *csv* to save the Data frame in CSV file.
- *haversine* is imported to calculate the distance between two coordinates.
- Selenium *webdriver* is used to interact with web applications and use their API.
- Selenium module *selenium.webdriver.common.by* is used with locators.
- Selenium module *selenium.webdriver.chrome.service* is used to initialize chrome driver.
- Selenium module *selenium.webdriver.support.wait* is used for explicit wait.
- *expected_conditions as EC* is used with explicit wait for initializing an expected condition.

2.2.Initialisation

Here, link of Asian Paints website’s store locator page is passed as well as the location of the chrome driver

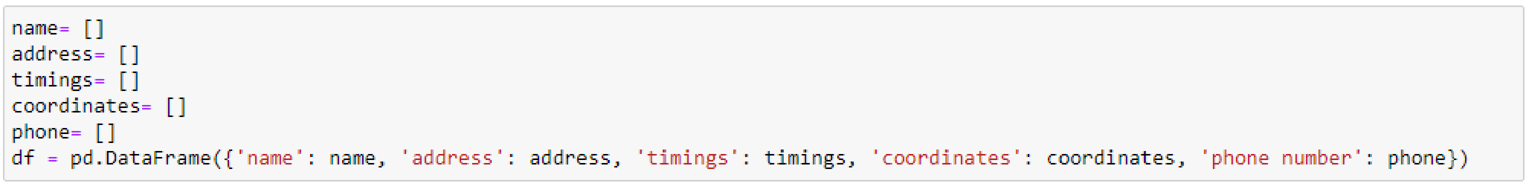


```
website = "https://www.asianpaints.com/store-locator.html"
path= Service('C:/Users/khale/Downloads/chromedriver_win32')
driver = webdriver.Chrome(service=path)
driver.get(website)
```

Fig: 2.2. Initialisation

2.3.Data structures used

To store the data of the store name, address, phone number, timings and coordinates, lists are used. And to store all the lists, a data frame is used.



```
name= []
address= []
timings= []
coordinates= []
phone= []
df = pd.DataFrame({'name': name, 'address': address, 'timings': timings, 'coordinates': coordinates, 'phone number': phone})
```

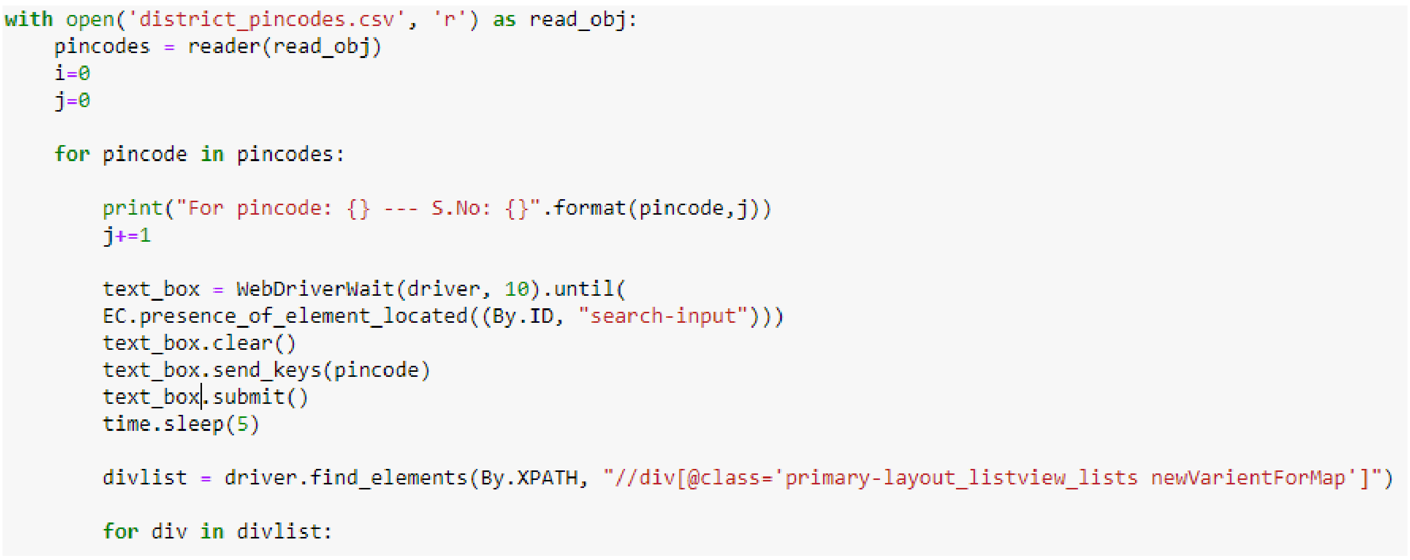
Fig: 2.3. Data structures used

- *name[]* is used to store names of the stores
- *address[]* is used to store the address of the stores
- *timings[]* is used to store the timings of the stores
- *coordinates[]* is used to store the latitude and longitude of the stores

- *phone[]* is used to store the phone number of the stores.
- Data Frame *df* is used to store the lists as a table

2.4.Main Body

Firstly, the file containing all the pin codes is opened, and all the pin codes are stored in the variable *pincodes*. Two variables *i*, *j* are used. *j* is used to show index number of the current pin code in used in the console log. The print statement shows the current pin code in use in the console log. A for loop is run iterating through each *pincode* in *pincodes.*



```
with open('district_pincodes.csv', 'r') as read_obj:
    pincodes = reader(read_obj)
    i=0
    j=0

    for pincode in pincodes:

        print("For pincode: {} --- S.No: {}".format(pincode,j))
        j+=1

        text_box = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "search-input")))
        text_box.clear()
        text_box.send_keys(pincode)
        text_box.submit()
        time.sleep(5)

        divlist = driver.find_elements(By.XPATH, "//div[@class='primary-layout_listview_lists newVarientForMap']")

        for div in divlist:
```

Fig: 2.4. Main body (1)

Another variable *text_box* is used to store the HTML element which takes pin codes as input. We use selenium explicit wait to wait if the presence of the element (*text_box*) is not located. *text_box* is searched by ID by using another selenium module (selenium.webdriver.common.by). After locating, the text inside is cleared and using the *send_keys* function, *pincode* is passed in the *text_box.*

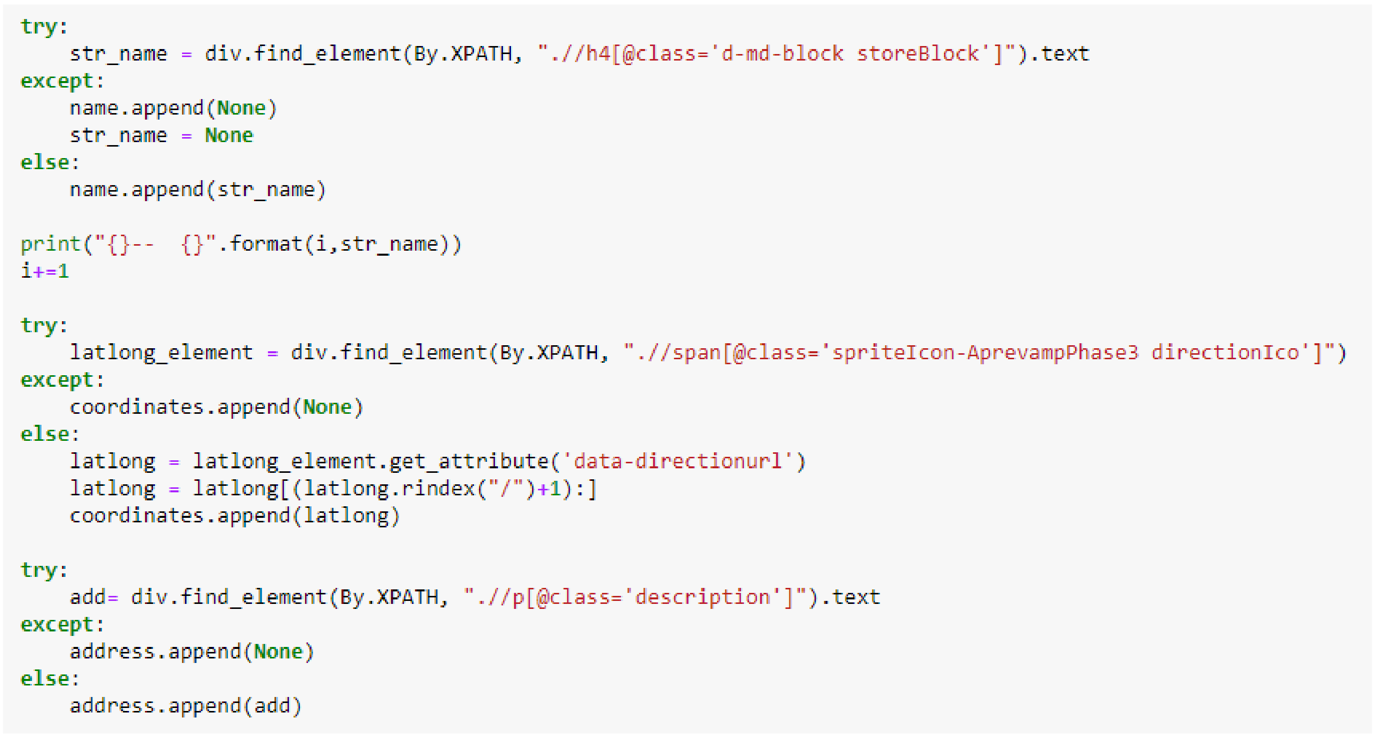
Time.sleep(5) forces the program to wait 5 seconds in order to let the page load.

List of all elements containing the store data are stored in variable *divlist*, which are searched using selenium xpath locator.

Another nested for loop is used to iterate through all the elements of the *divlist.*

While iterating through each *div* element in the *divlist*, which contains all the data for a single store, we use selenium locators to find for elements which contain the useful information.

- For name, xpath is used to search for element containing name and then text method is used to get the store name. Store name is then appended to the name list

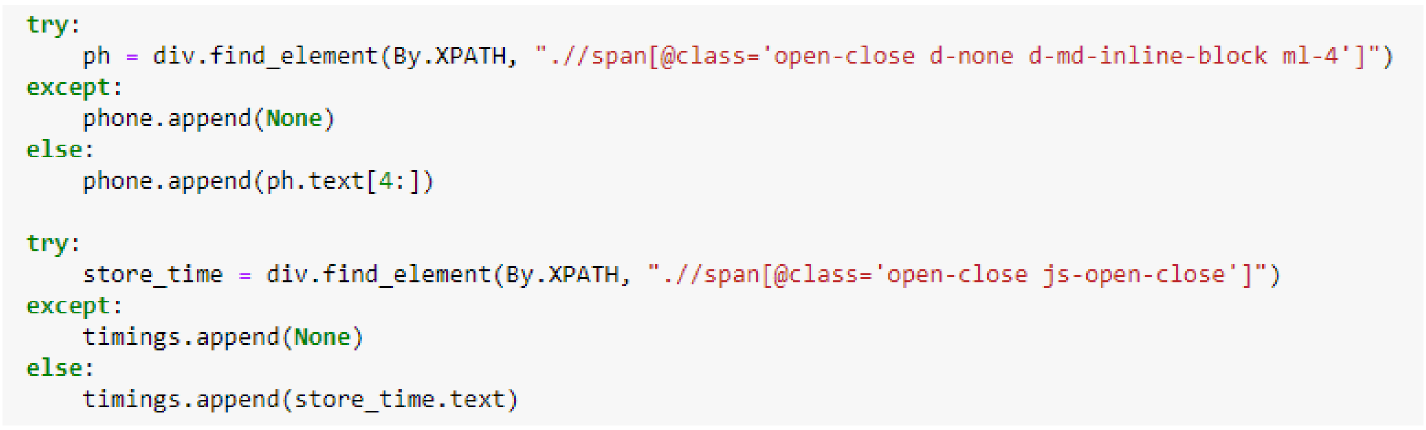


```
try:
    str_name = div.find_element(By.XPATH, ".//h4[@class='d-md-block storeBlock']").text
except:
    name.append(None)
    str_name = None
else:
    name.append(str_name)

print("{}--  {}".format(i,str_name))
i+=1

try:
    latlong_element = div.find_element(By.XPATH, ".//span[@class='spriteIcon-AprevampPhase3 directionIco']")
except:
    coordinates.append(None)
else:
    latlong = latlong_element.get_attribute('data-directionurl')
    latlong = latlong[(latlong.rindex("/")+1):]
    coordinates.append(latlong)

try:
    add= div.find_element(By.XPATH, ".//p[@class='description']").text
except:
    address.append(None)
else:
    address.append(add)
```

Fig: 2.5. Main body (2)

- For latitude and longitude xpath is used to search for the element containing location. Since the coordinates of the store are passed in a link which redirect to google maps, the value of that link is retrieved with the help of *get_attribute* method and the string is stored in a variable *latlong*. By using indexing the desired coordinates are achieved.

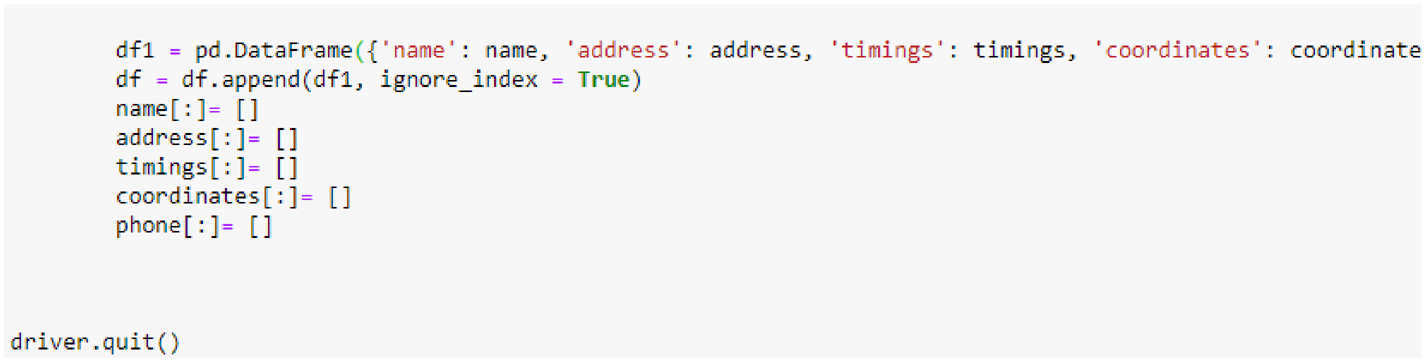
- For address, phone number, timings, same procedure is followed. Xpath is used to locate the elements and the text method is used to retrieve the value. For phone number, indexing is used to get the 10 digit phone number only. All the values are appended into their respected lists



```
try:
    ph = div.find_element(By.XPATH, ".//span[@class='open-close d-none d-md-inline-block ml-4']")
except:
    phone.append(None)
else:
    phone.append(ph.text[4:])

try:
    store_time = div.find_element(By.XPATH, ".//span[@class='open-close js-open-close']")
except:
    timings.append(None)
else:
    timings.append(store_time.text)
```

Fig: 2.6. Main Body (3)

All *find_element* are in *try and except* statements because if any one of the elements is not present, *None* value should be appended in that particular list, and the loop should carry on instead of throwing an error.



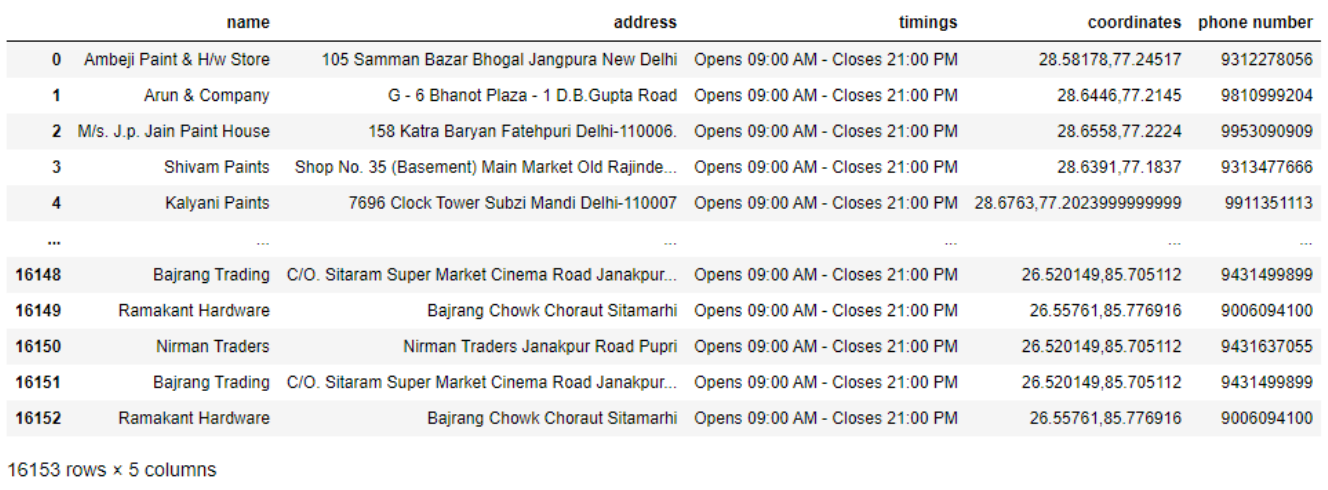
```
        df1 = pd.DataFrame({'name': name, 'address': address, 'timings': timings, 'coordinates': coordinate
        df = df.append(df1, ignore_index = True)
        name[:]= []
        address[:]= []
        timings[:]= []
        coordinates[:]= []
        phone[:]= []



driver.quit()
```

Fig: 2.7. Main body (4)

A data frame *df1* is declared outside of the inner loop. In *df1* all the data of all the stores in that particular pin code is stored. This data frame df1 is the appended to the main data frame *df*. Now, all the list are then emptied before iterating to the next pin code.

When the program has iterated through both the loops completely, *driver.quit()* is executed which closes the chrome tab.

2.5.Using Data Frame

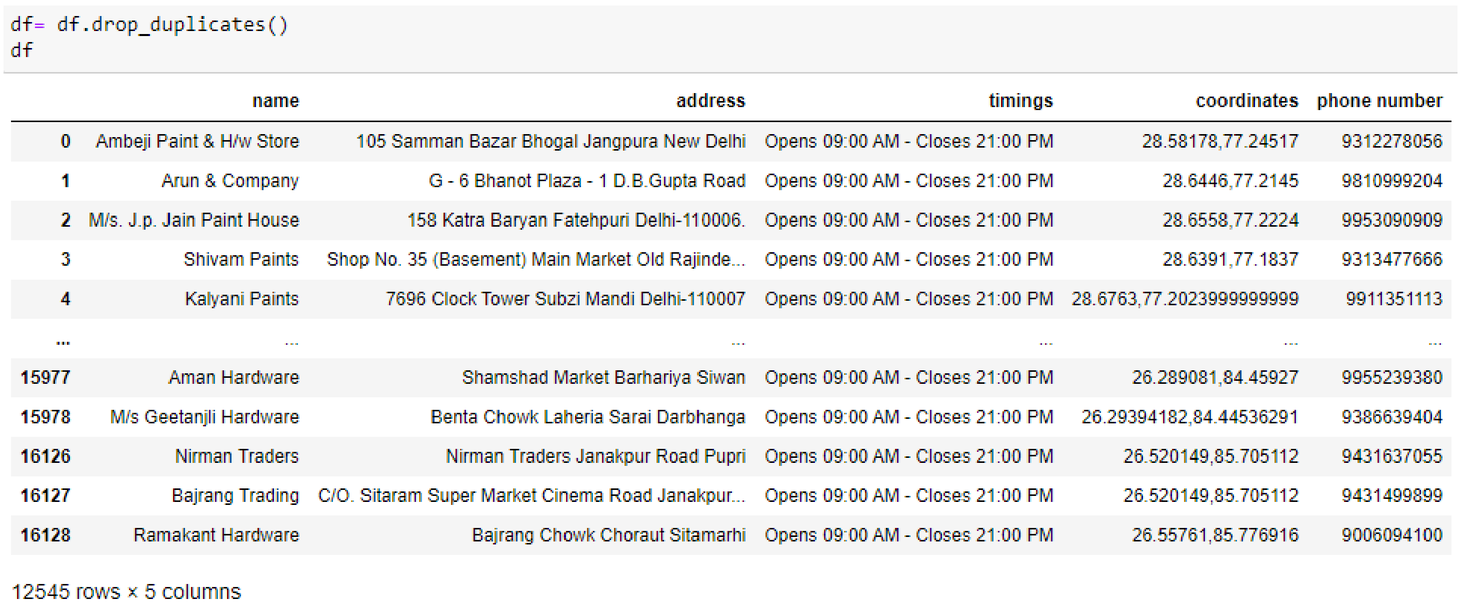


	name	address	timings	coordinates	phone number
0	Ambeji Paint & H/w Store	105 Samman Bazar Bhogal Jangpura New Delhi	Opens 09:00 AM - Closes 21:00 PM	28.58178,77.24517	9312278056
1	Arun & Company	G - 6 Bhanot Plaza - 1 D.B.Gupta Road	Opens 09:00 AM - Closes 21:00 PM	28.6446,77.2145	9810999204
2	M/s. J.p. Jain Paint House	158 Katra Baryan Fatehpuri Delhi-110006.	Opens 09:00 AM - Closes 21:00 PM	28.6558,77.2224	9953090909
3	Shivam Paints	Shop No. 35 (Basement) Main Market Old Rajinde...	Opens 09:00 AM - Closes 21:00 PM	28.6391,77.1837	9313477666
4	Kalyani Paints	7696 Clock Tower Subzi Mandi Delhi-110007	Opens 09:00 AM - Closes 21:00 PM	28.6763,77.2023999999999	9911351113
...	...	...	...	...	...
16148	Bajrang Trading	C/O. Sitaram Super Market Cinema Road Janakpur...	Opens 09:00 AM - Closes 21:00 PM	26.520149,85.705112	9431499899
16149	Ramakant Hardware	Bajrang Chowk Choraut Sitamarhi	Opens 09:00 AM - Closes 21:00 PM	26.55761,85.776916	9006094100
16150	Nirman Traders	Nirman Traders Janakpur Road Pupri	Opens 09:00 AM - Closes 21:00 PM	26.520149,85.705112	9431637055
16151	Bajrang Trading	C/O. Sitaram Super Market Cinema Road Janakpur...	Opens 09:00 AM - Closes 21:00 PM	26.520149,85.705112	9431499899
16152	Ramakant Hardware	Bajrang Chowk Choraut Sitamarhi	Opens 09:00 AM - Closes 21:00 PM	26.55761,85.776916	9006094100

16153 rows × 5 columns

Fig: 2.2. Data Frames

All the data extracted is stored in the Data Frame *df*. In the above figure, a total of 16153 rows of data is scraped from the website i.e., total of 16152 stores are retrieved. But some of the stores may be common for more than one pin codes, therefore all the duplicate data must be removed. This is done using Pandas *delete_duplicate* method.



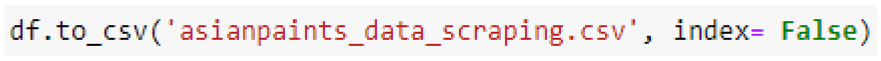
```
df= df.drop_duplicates()
df
```

	name	address	timings	coordinates	phone number
0	Ambeji Paint & H/w Store	105 Samman Bazar Bhogal Jangpura New Delhi	Opens 09:00 AM - Closes 21:00 PM	28.58178,77.24517	9312278056
1	Arun & Company	G - 6 Bhanot Plaza - 1 D.B.Gupta Road	Opens 09:00 AM - Closes 21:00 PM	28.6446,77.2145	9810999204
2	M/s. J.p. Jain Paint House	158 Katra Baryan Fatehpuri Delhi-110006.	Opens 09:00 AM - Closes 21:00 PM	28.6558,77.2224	9953090909
3	Shivam Paints	Shop No. 35 (Basement) Main Market Old Rajinde...	Opens 09:00 AM - Closes 21:00 PM	28.6391,77.1837	9313477666
4	Kalyani Paints	7696 Clock Tower Subzi Mandi Delhi-110007	Opens 09:00 AM - Closes 21:00 PM	28.6763,77.2023999999999	9911351113
...	...	...	...	...	...
15977	Aman Hardware	Shamshad Market Barhariya Siwan	Opens 09:00 AM - Closes 21:00 PM	26.289081,84.45927	9955239380
15978	M/s Geetanjli Hardware	Benta Chowk Laheria Sarai Darbhanga	Opens 09:00 AM - Closes 21:00 PM	26.29394182,84.44536291	9386639404
16126	Nirman Traders	Nirman Traders Janakpur Road Pupri	Opens 09:00 AM - Closes 21:00 PM	26.520149,85.705112	9431637055
16127	Bajrang Trading	C/O. Sitaram Super Market Cinema Road Janakpur...	Opens 09:00 AM - Closes 21:00 PM	26.520149,85.705112	9431499899
16128	Ramakant Hardware	Bajrang Chowk Choraut Sitamarhi	Opens 09:00 AM - Closes 21:00 PM	26.55761,85.776916	9006094100

12545 rows × 5 columns

Fig: 2.9. Delete Duplicate

After deleting the duplicates, there are only 12545 rows of data, therefore there were multiple stores common for the same pin code. The data in Data Frame is then stored in a csv file.



```
df.to_csv('asianpaints_data_scraping.csv', index= False)
```

Fig: 2.10. Save to CSV

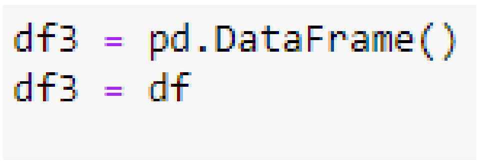
Chapter 3

Analysis: Filtering Nearest Stores

After the data is stored in the Data Frame, some analysis can be performed and some useful insights can be taken out from the raw data.

After scraping the useful data, the stores which are nearest to the user can be calculated. The coordinates of the user are entered and the distance is calculated from each store. The table of store details is displayed sorted in ascending order of the distance. This location-based filtering method is also used in mobile or web applications to show nearest options.

3.1.New Data Frame initialization



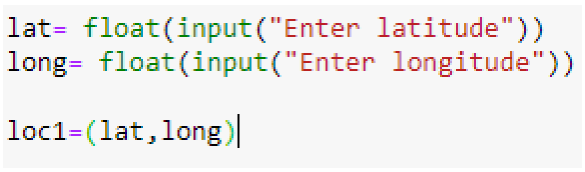
```
df3 = pd.DataFrame()
df3 = df
```

Fig 3.1. *Initializing new data frame*

A new Data Frame (*df3*) is initialized for data analysis. *df3* will have an extra column for storing distance from the user entered coordinates to the coordinates of the store

3.2.User entered coordinates

The user entered latitude and longitude is stored in another variable *loc1* as a tuple.

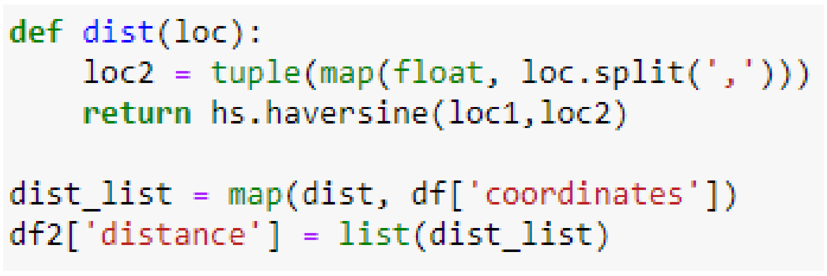


```
lat= float(input("Enter latitude"))
long= float(input("Enter longitude"))

loc1=(lat,long)
```

Fig 3.2. User entered coordinates

3.3.Calculating distance

The distance between every store and the user coordinates in calculated and stored in the new column of *df3, 'distance'*. For this *map()* function is used and a function *dist()* and an iterable (*'coordinates'* column of *df*) is passed in it. Function *dist()* is used to convert the string coordinates to float and store them as tuples and then calculates the distance between the two coordinates using haversine library.

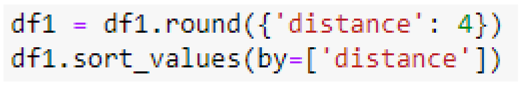


```
def dist(loc):
    loc2 = tuple(map(float, loc.split(',')))
    return hs.haversine(loc1,loc2)

dist_list = map(dist, df['coordinates'])
df2['distance'] = list(dist_list)
```

Fig 3.3. Calculating distance

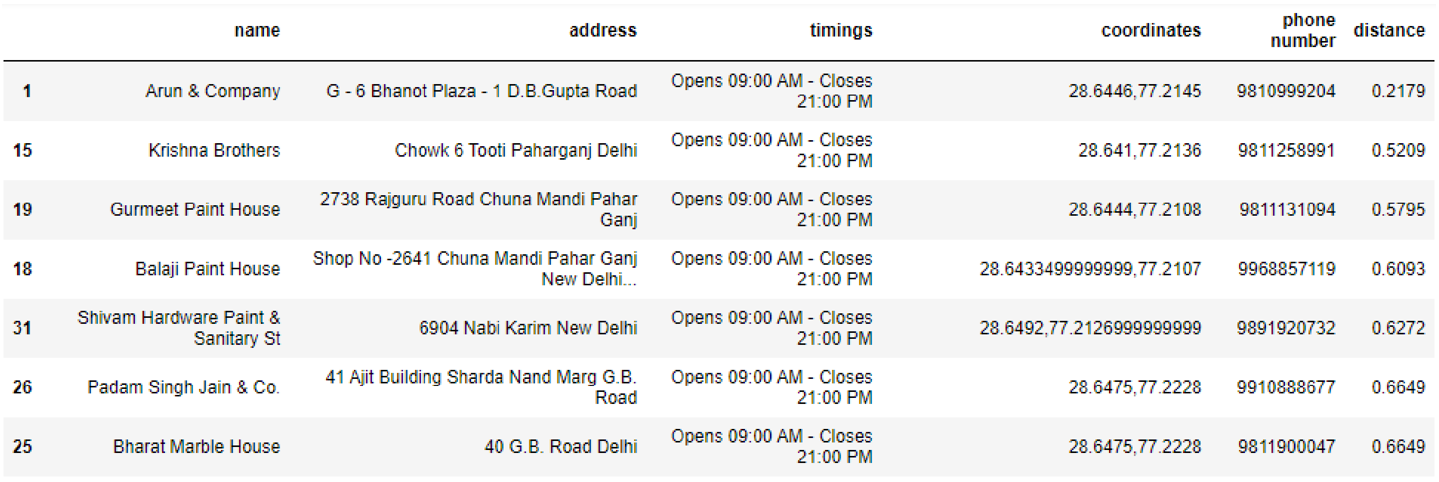
3.4.Sorting and displaying nearest stores

The values of *df3['distance']* are then rounded to nearest 4 decimals using *round()* method. The round of values are then sorted according to increasing distance by using *sort_values()* method and the top 10 records are shown. Hence, the nearest 10 stores of Asian Paints to the user are displayed. In the figure below, user entered coordinates are (28.644800,77.216721)



```
df1 = df1.round({'distance': 4})
df1.sort_values(by=['distance'])
```

Fig 3.4. Sorting



	name	address	timings	coordinates	phone number	distance
1	Arun & Company	G - 6 Bhanot Plaza - 1 D.B.Gupta Road	Opens 09:00 AM - Closes 21:00 PM	28.6446,77.2145	9810999204	0.2179
15	Krishna Brothers	Chowk 6 Tooti Paharganj Delhi	Opens 09:00 AM - Closes 21:00 PM	28.641,77.2136	9811258991	0.5209
19	Gurmeet Paint House	2738 Rajguru Road Chuna Mandi Pahar Ganj	Opens 09:00 AM - Closes 21:00 PM	28.6444,77.2108	9811131094	0.5795
18	Balaji Paint House	Shop No -2641 Chuna Mandi Pahar Ganj New Delhi...	Opens 09:00 AM - Closes 21:00 PM	28.6433499999999,77.2107	9968857119	0.6093
31	Shivam Hardware Paint & Sanitary St	6904 Nabi Karim New Delhi	Opens 09:00 AM - Closes 21:00 PM	28.6492,77.2126999999999	9891920732	0.6272
26	Padam Singh Jain & Co.	41 Ajit Building Sharda Nand Marg G.B. Road	Opens 09:00 AM - Closes 21:00 PM	28.6475,77.2228	9910888677	0.6649
25	Bharat Marble House	40 G.B. Road Delhi	Opens 09:00 AM - Closes 21:00 PM	28.6475,77.2228	9811900047	0.6649

Fig 3.5. Nearest stores

Chapter 4

Challenges and Suggestive Measures

In this assignment, data of Asian Paints website is scraped using Python and Selenium. The data (store name, store address, store phone number, store timings, store coordinates) of all the stores of Asian Paints in India is scraped using web scraping techniques and the stored in a CSV file.

4.1. Challenges faced

The following challenges were encountered while scraping the data:

4.1.1. Dynamic mechanism needed to extract data from all the pin codes.

The Asian Paints website displays all the stores of a single pin code at a time. Some dynamic mechanism was needed to enter pin code one by one and extract data from all the pin codes.

4.1.2. Selenium extracting data of previous pin code instead of waiting for the page to refresh.

While iterating through pin code, when the next pin code is entered, implicit wait or explicit wait can be used to let the page reload and then proceed with the code. Explicit wait can be used with the expected condition as *staleness_of* an element. But since the elements (div) containing the store detail were present for the previous pin code as well, therefore as soon as the new pin code is entered, selenium locates the presence of the previous pin code's elements and hence proceed to extract their data again without waiting for the page to completely refresh and the elements containing the data of the current pin code to appear. This is because explicit wait and implicit wait conditions work in such a way that it will wait for maximum specified time or until the presence of the elements is located. Using explicit wait with *staleness_of* method would give a timeout error as the div would never 'stale' as it was present in the previous pin code as well.

4.1.3. The program may throw an error due to increasing size of the lists.

All the store date are stored in 5 lists. The lists may become extremely large depending upon the amount of data extracted from the website. Due to this, it may encounter an error.

4.2. Suggestive measures

As per the challenges faced, the following measures were taken:

4.2.1. Selenium is used as a dynamic mechanism.

Selenium, a web automation technique, is used to locate the pin code text input with the help of locators, and input one pin code at a time from a CSV containing all the pin codes of all the districts in India.

4.2.2. *time.sleep()* is used to let the page refresh.

Since, implicit and explicit waits cannot be used, *time.sleep(5)* is used, which is usually discouraged as it slows down the program and increases the time complexity un necessarily. Here '*5*' indicates, stop the program for 5 seconds to let the page load.

4.2.3. Data of each pin code is stored in a Data Frame to minimize the length of the lists.

After retrieving the data from all the stores in a particular pin code, the data in the lists is appended to a Data frame (*df*) and then the lists are emptied before iterating to another pin code, hence the lists will not get excessively long.

Chapter 5

Conclusion

Hence, by using both Pandas and Selenium, the data from the Asian Paints website is scraped. This program can be used to scrape data from almost all the dynamic websites by locating the input fields and retrieving the elements in which the data is stored.

5.1. Limitations

The major limitation of this program is its complexity. Its time complexity is $O(n^2)$ as there are two nested loops.

Also, Selenium in Python is comparatively slower, than that used with Java. This also increases the time of the completion of the program.

Currently, pin codes of all the districts are being used as the time duration of the completion of the code is almost 4 hours. Hence, some of the stores which are not under those pin codes might not be retrieved.

5.2. Further work and Improvements

Improvements in the code can be done by using a faster library for web automation, which will help in increasing the efficiency of the program.

For further work, with the increase in the efficiency of the program, all the pin codes of India can be used to retrieve all the data from all the stores so no stores are left.