Optional_task

September 26, 2021

1 MapUp @ Data Analyst; Python mandatory task

- Candidate name :`Nijatullah Mansoor`

1.1 Task

Using python, scrape the toll data. If you have been using selenium till now,it might not be useful since rates vary frequently. Explore the xhr requests of the webpage

Here is the website link we want you to scrap. The website gives toll rates for different entry and exits on I95 and I495 express lanes in US. Toll rates vary every few minutes depending on the congestion of the road. Click here.

You'll find a json file from where these toll rates are getting updated. Clean the json file and transform it to a df

Load the data into a csv with all the entry and exit combinations along with the toll rates at that moment. A sample csv is given for your reference

Setup a pipeline which would repeat all the steps when required

Send us your python and csv files for this task by Monday 9 AM. You can contact us if you have any queries

the final Dataset should looks like this.

Let's import the rerquired library.

[34]:
```
import pandas as pd # for data manipulation and analysis
import requests # The requests library is the de facto standard for making HTTP␣
 ↪requests in Python.
import time # to keep track of the whole process

from selenium import webdriver
from selenium.webdriver.support.ui import Select
```

[2]:
```
df = pd.read_csv('OD_combinations.csv')
df.head()
```

[2]:
```
  entry_id exit_id                    entry_label  \
0    202NO   224ND   495 Express Lanes/I-495/I-95
1    202NO   223ND   495 Express Lanes/I-495/I-95
```

```
2    202NO   222ND   495 Express Lanes/I-495/I-95
3    202NO   201ND   495 Express Lanes/I-495/I-95
4    203NO   224ND  Old Keene Mill Road/Route 644

                     exit_label      ods  path Direction status  price  \
0                Washington D.C.  od_1265    95        NB   open    0.8
1           Pentagon/Eads Street  od_1264    95        NB   open    1.0
2  Seminary Road NB (HOV-3 ONLY)  od_1263    95        NB   open    0.0
3         I-395 Near Edsall Road  od_1146    95        NB   open    0.2
4                Washington D.C.  od_1262    95        NB   open    0.6

              time
0  9/21/2021 15:43
1  9/21/2021 15:43
2  9/21/2021 15:43
3  9/21/2021 15:43
4  9/21/2021 15:43
```

The website gives toll rates for different entry and exits on I95 and I495 express lanes in US. later we will get data fro different entry and exits point let's first get for one entrey and exit point.

Lets get data in `Northbond` of entry point is `Jones Branch Drive/Route 123` and exit point is `495 Express End (near MD)`.

Now we will select manualy the direction of travaling and the entry point and the exit point later with the help of selenium we will select dynamically.

`website url`

url = 'https://www.expresslanes.com/map-your-trip'

let's define the header.

[4]:
```
user_agent = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36␣
 ↪(KHTML, like Gecko) Chrome/93.0.4577.82 Safari/537.36'
cookie =␣
 ↪'visid_incap_1690171=pj6EUeOOTJ+jxMZ4HDrWeraETGEAAAAAQUIPAAAAAACItBVkc7A8RILyPfYu+p0g;
 ↪ _ga=GA1.2.294496981.1632404687; _gid=GA1.2.54831270.1632404687;␣
 ↪SSESS95f4477c0936cd6d99fc5fd63b07d31f=zRhmibPPJ0aofpvgY9Y7XLwnbbwUtO-1vHBhbwfHa3s;
 ↪ nlbi_1690171=dvf+JLChIAX3SOZwP9T7GgAAAABpI7QwRHCg3+JVnY05F+Pt;␣
 ↪incap_ses_708_1690171=+5olY9guniyV8MzSsVLTCWb1T2EAAAAAOsX61DGiRxw6oMtg4qoJGw=='
accept_language = 'en-GB,en-US;q=0.9,en;q=0.8,lb;q=0.7,ps;q=0.6,kn;q=0.5'
authority = 'www.expresslanes.com'
x_requeste = 'XMLHttpRequest'
method = 'GET'


headers = {'User-Agent': user_agent,
           'cookie':cookie,
           'accept-language':accept_language,
```

```
            'x-requested-with':x_requeste,
            'method':method,
            'authority':authority,
           }
```

1.1.1 Requested URL

[5]:
```
url = 'https://www.expresslanes.com/maps-api/infra-price-confirmed-all'
response = requests.post(url,headers=headers)
```

[6]:
```
response
```

[6]:
```
<Response [200]>
```

[7]:
```
type(response)
```

[7]:
```
requests.models.Response
```

1.2 Let's get json file

[8]:
```
json_data = response.json()
json_data.keys()
```

[8]:
```
dict_keys(['error', 'error_text', 'response', 'direction_95', '#cache'])
```

our Data is exitsin the `response` key.

[9]:
```
df = json_data['response']
```

[10]:
```
type(df)
```

[10]:
```
list
```

[11]:
```
df[0]
```

[11]:
```
{'od': 'od_1024',
 'price': '0.85',
 'road': '495',
 'ratetype': 'DTA',
 'time': '2021-09-26 07:50:39',
 'direction': 'N',
 'status': 'open'}
```

the list contain a dictionary object.

[12]:
```
print(df[0]['od'])
print(df[0]['price'])
```

```
print(df[0]['road'])
print(df[0]['ratetype'])
print(df[0]['time'])
print(df[0]['direction'])
print(df[0]['status'])
```

```
od_1024
0.85
495
DTA
2021-09-26 07:50:39
N
open
```

1.3 Let's create a empty list

[24]:
```
entry_id = []
exit_id = []
ods = []
prices = []
roads = []
ratetype = []
dates = []
direction = []
status = []
entry_label = []
exit_label = []
```

let's loop throug and get the data.

[30]:
```
for k in range(len(df)):
    ods.append(df[k]['od'])
    prices.append(df[k]['price'])
    roads.append(df[k]['road'])
    ratetype.append(df[k]['ratetype'])
    direction.append(df[k]['direction'])
    status.append(df[k]['status'])
    dates.append(df[k]['time'])
    entry_label.append('Jones Branch Drive/Route 123') # we will get this now␣
↪manually later with the help fo selenium we will get daynamically
    exit_label.append('495 Express End (near MD)') # same goes here
    entry_id.append('202NO') # same here
    exit_id.append('224ND') # same here
```

Let's create an empty dataFrame

[31]:
```
final_df = pd.
↪DataFrame(columns=['Entry_id','Exit_id','Entry_label','Exit_label','ods','path','direction'
```

[32]:
```
final_df['Entry_id'] = entry_id
final_df['Exit_id'] = exit_id
final_df['ods'] = ods
final_df['Entry_label'] = entry_label
final_df['Exit_label'] = exit_label
final_df['status'] = status
final_df['path'] = roads
final_df['price'] = prices
final_df['direction'] = direction
final_df['Date'] = dates
final_df['ratetype'] = ratetype
```

[33]:
```
final_df.head()
```

[33]:

	Entry_id	Exit_id	Entry_label	Exit_label
0	202NO	224ND	Jones Branch Drive/Route 123	495 Express End (near MD)
1	202NO	224ND	Jones Branch Drive/Route 123	495 Express End (near MD)
2	202NO	224ND	Jones Branch Drive/Route 123	495 Express End (near MD)
3	202NO	224ND	Jones Branch Drive/Route 123	495 Express End (near MD)
4	202NO	224ND	Jones Branch Drive/Route 123	495 Express End (near MD)

	ods	path	direction	status	price	Date	ratetype
0	od_1024	495	N	open	0.85	2021-09-26 07:50:39	DTA
1	od_1025	495	N	open	0.85	2021-09-26 07:49:22	DTA
2	od_1026	495	N	open	0.90	2021-09-26 07:50:39	DTA
3	od_1027	495	N	open	1.10	2021-09-26 07:49:22	DTA
4	od_1028	495	N	open	1.30	2021-09-26 07:49:22	DTA

Now let's get data of multiple entry and exit points of `Northbond` and `southbound`. I will create a robot which is capable of Select the direction of traveling then click on the access points to see detailed maps of Express Lanes entries and exits. Next Choose Northbound entry and exit points. and finally click on `View your Rout` button. And get a final dataset which we can convert into `.csv` ,`.xlsx` or we can store it in `MySQL database`.

from the `Northbound` the entry point will be `Braddock Road` and all its corresponding exit points. from the `Southbound` the entry point will be `Route 7 (Leesburg Pike)` it's corresponding exit points.

1.3.1 piplene to repeat the above task.

[61]:
```
start = time.time() # to keep track of time.

user_agent = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36␣
↪(KHTML, like Gecko) Chrome/93.0.4577.82 Safari/537.36'
```

```
cookie =␣
 ↪'visid_incap_1690171=pj6EUeOOTJ+jxMZ4HDrWeraETGEAAAAAQUIPAAAAAACItBVkc7A8RILyPfYu+p0g;
 ↪ _ga=GA1.2.294496981.1632404687; _gid=GA1.2.54831270.1632404687;␣
 ↪SSESS95f4477c0936cd6d99fc5fd63b07d31f=zRhmibPPJ0aofpvgY9Y7XLwnbbwUtO-1vHBhbwfHa3s;
 ↪ nlbi_1690171=dvf+JLChIAX3SOZwP9T7GgAAAABpI7QwRHCg3+JVnY05F+Pt;␣
 ↪incap_ses_708_1690171=+5olY9guniyV8MzSsVLTCWb1T2EAAAAAOsX61DGiRxw6oMtg4qoJGw=='
accept_language = 'en-GB,en-US;q=0.9,en;q=0.8,lb;q=0.7,ps;q=0.6,kn;q=0.5'
authority = 'www.expresslanes.com'
x_requeste = 'XMLHttpRequest'
method = 'GET'

headers = {'User-Agent': user_agent,
           'cookie':cookie,
           'accept-language':accept_language,
           'x-requested-with':x_requeste,
           'method':method,
           'authority':authority,
          }

path = r'C:\Users\nijat\Desktop/Data Science/Preparation For Interview/MapUP/
 ↪Optional_task/chromedriver'
driver = webdriver.Chrome(executable_path=path)

# url for the website
url = 'https://www.expresslanes.com/map-your-trip'
driver.get(url)

time.sleep(3) # waite for 3 second to load the webpage

# let's define empty list to store data
entry_id = []
exit_id = []
ods = []
prices = []
roads = []
ratetype = []
dates = []
direction = []
status = []
entry_label = []
exit_label = []


for i in range(1,3): # two iteration loop one time for Northbound and second␣
 ↪time for Southbound
    select_direction  = Select(driver.find_element_by_id('DirectionSelect'))
```

```
   select_direction.select_by_index(i) # index 1 is for Northbound and index 2␣
↪is for southbound

   # Now let's chose the entry and exit point.

   # Entry point
   time.sleep(2)
   select_entry = Select(driver.find_element_by_id('EntrySelect'))
   select_entry.select_by_index(5)
   element = select_entry.first_selected_option
   entry_point_label = element.text
   entry_point_id = element.get_attribute('value')

   # for one entry point there are many exit points so we will use another␣
↪nested loop to select the exit points

   for j in range(1,8): # each entry point has 7 exit points. so we will use 7␣
↪iterations.

       time.sleep(2) # to load the webpage
       select_exit = Select(driver.find_element_by_id("ExitSelect"))
       select_exit.select_by_index(j)
       element = select_exit.first_selected_option
       exit_point_label = element.text
       exit_point_id = element.get_attribute('value')


       # now click of the button and get the json file
       # click of view round button
       time.sleep(3) # we will give some time to load everything properly
       driver.find_element_by_id('ViewRouteButton').click()
       response = requests.post('https://www.expresslanes.com/maps-api/
↪infra-price-confirmed-all',headers=headers) # requested URL
       json_data = response.json()
       df = json_data['response']

       # now let's loop throug the df to get data.

       for k in range(len(df)):
           entry_id.append(entry_point_id)
           exit_id.append(exit_point_id)
           ods.append(df[k]['od'])
           prices.append(df[k]['price'])
           roads.append(df[k]['road'])
           ratetype.append(df[k]['ratetype'])
           direction.append(df[k]['direction'])
           status.append(df[k]['status'])
```

```
            dates.append(df[k]['time'])
            entry_label.append(entry_point_label)
            exit_label.append(exit_point_label)

# initialize empty dataframe.
final_df = pd.
 ↪DataFrame(columns=['Entry_id','Exit_id','Entry_label','Exit_label','ods','path','direction'
final_df['Entry_id'] = entry_id
final_df['Exit_id'] = exit_id
final_df['ods'] = ods
final_df['Entry_label'] = entry_label
final_df['Exit_label'] = exit_label
final_df['status'] = status
final_df['path'] = roads
final_df['price'] = prices
final_df['direction'] = direction
final_df['Date'] = dates
final_df['ratetype'] = ratetype
# the require time to complete the process
end = time.time()
print ("Time Taken for the complete process is:{} seconds".format((end-start)))
# quitting the driver (browser)
driver.quit()
# returning the dataframe formed
```

```
Time Taken for the complete process is:103.57664585113525 seconds
```

[62]:
```
final_df
```

[62]:
```
     Entry_id Exit_id              Entry_label                 Exit_label  \
0       190NO   181ND           Braddock Road  495 Express End (near MD)
1       190NO   181ND           Braddock Road  495 Express End (near MD)
2       190NO   181ND           Braddock Road  495 Express End (near MD)
3       190NO   181ND           Braddock Road  495 Express End (near MD)
4       190NO   181ND           Braddock Road  495 Express End (near MD)
…         …      …                      …                         …
3817    186SO  2239ND  Route 7 (Leesburg Pike)       Pentagon/Eads Street
3818    186SO  2239ND  Route 7 (Leesburg Pike)       Pentagon/Eads Street
3819    186SO  2239ND  Route 7 (Leesburg Pike)       Pentagon/Eads Street
3820    186SO  2239ND  Route 7 (Leesburg Pike)       Pentagon/Eads Street
3821    186SO  2239ND  Route 7 (Leesburg Pike)       Pentagon/Eads Street

          ods path direction status price                 Date ratetype
0     od_1024  495         N   open  0.85  2021-09-26 11:30:36      DTA
1     od_1025  495         N   open  0.90  2021-09-26 11:29:21      DTA
2     od_1026  495         N   open  1.10  2021-09-26 11:30:36      DTA
3     od_1027  495         N   open  1.30  2021-09-26 11:29:21      DTA
```

```
4     od_1028  495         N   open  1.50  2021-09-26 11:29:21      DTA
…       …  …       …     …   …                    …       …
3817  od_1019  495         N   open  1.85  2021-09-26 11:30:39      DTA
3818  od_1020  495         N   open  2.05  2021-09-26 11:29:19      DTA
3819  od_1021  495         N   open  2.25  2021-09-26 11:29:19      DTA
3820  od_1022  495         N   open  2.45  2021-09-26 11:30:39      DTA
3821  od_1023  495         N   open  0.85  2021-09-26 11:29:21      DTA

[3822 rows x 11 columns]
```

[63]:
```
final_df['Entry_id'].unique()
```

[63]:
```
array(['190NO', '186SO'], dtype=object)
```

[64]:
```
final_df['Exit_id'].unique()
```

[64]:
```
array(['181ND', '182ND', '183ND', '185ND', '186ND', '187ND', '188ND',
       '187SD', '189SD', '190SD', '191SD', '192SD', '2249ND', '2239ND'],
      dtype=object)
```

1.4 Now let's convert this DataFrame to a csv file.

[65]:
```
final_df.to_csv('toll_data.csv',index=False)
```

Let's read csv file back.

[67]:
```
df = pd.read_csv('toll_data.csv')
```

[68]:
```
df.head()
```

[68]:
```
  Entry_id Exit_id    Entry_label                 Exit_label      ods  path  \
0    190NO   181ND  Braddock Road  495 Express End (near MD)  od_1024   495
1    190NO   181ND  Braddock Road  495 Express End (near MD)  od_1025   495
2    190NO   181ND  Braddock Road  495 Express End (near MD)  od_1026   495
3    190NO   181ND  Braddock Road  495 Express End (near MD)  od_1027   495
4    190NO   181ND  Braddock Road  495 Express End (near MD)  od_1028   495

  direction status  price                 Date ratetype
0         N   open   0.85  2021-09-26 11:30:36      DTA
1         N   open   0.90  2021-09-26 11:29:21      DTA
2         N   open   1.10  2021-09-26 11:30:36      DTA
3         N   open   1.30  2021-09-26 11:29:21      DTA
4         N   open   1.50  2021-09-26 11:29:21      DTA
```

[69]:
```
df.tail()
```

[69]:
```
     Entry_id Exit_id              Entry_label            Exit_label      ods  \
3817    186SO  2239ND  Route 7 (Leesburg Pike)  Pentagon/Eads Street  od_1019
3818    186SO  2239ND  Route 7 (Leesburg Pike)  Pentagon/Eads Street  od_1020
3819    186SO  2239ND  Route 7 (Leesburg Pike)  Pentagon/Eads Street  od_1021
3820    186SO  2239ND  Route 7 (Leesburg Pike)  Pentagon/Eads Street  od_1022
3821    186SO  2239ND  Route 7 (Leesburg Pike)  Pentagon/Eads Street  od_1023

      path direction status  price                 Date ratetype
3817   495         N   open   1.85  2021-09-26 11:30:39      DTA
3818   495         N   open   2.05  2021-09-26 11:29:19      DTA
3819   495         N   open   2.25  2021-09-26 11:29:19      DTA
3820   495         N   open   2.45  2021-09-26 11:30:39      DTA
3821   495         N   open   0.85  2021-09-26 11:29:21      DTA
```

similarly we can convert this to `xlsx` file or we can directly store this data in `MySQL` database.

[72]:
```
final_df.to_excel('toll_data.xlsx',index=False)
```

1.5 Let's store this data in `MySQL` database.

[73]:
```
#Import MySQL connector module
import mysql.connector
from sqlalchemy import create_engine
import pandas as pd

tableName = 'toll_data'


try:
    engine = create_engine("mysql://root:nijat123@localhost/mydb")
    connection = engine.connect()
    final_df.
 ↪to_sql(name=tableName,con=connection,if_exists='replace',index=False)

except mysql.connector.Error as e:
    print("Error writting data to MySQL table", e)

except ValueError as vx:
    print(vx)

except Exception as ex:
    print(ex)

else:
    print("Table %s created successfully."%tableName)

finally:
```

```
    if not connection.closed:
        connection.close()
        print("MySQL connection is closed")
```

```
Table toll_data created successfully.
MySQL connection is closed
```

1.6 Reading data from MySQL database table into pandas dataframe

[74]:
```
#Import MySQL connector module
import mysql.connector
from sqlalchemy import create_engine
import pandas as pd

tableName = 'route_123_495_express'


try:
    engine = create_engine("mysql://root:nijat123@localhost/mydb")
    connection = engine.connect()
    qury = "select * from toll_data;"
    df = pd.read_sql(qury,connection)


except mysql.connector.Error as e:
    print("Error reading from MySQl database.", e)

except ValueError as vx:
    print(vx)

except Exception as ex:
    print(ex)

else:
    print("data has been read successfully.")

finally:
    if not connection.closed:
        connection.close()
        print("MySQL connection is closed")
```

```
data has been read successfully.
MySQL connection is closed
```

[75]:
```
df.head()
```

[75]:

	Entry_id	Exit_id	Entry_label	Exit_label	ods path	\
0	190NO	181ND	Braddock Road 495	Express End (near MD)	od_1024	495
1	190NO	181ND	Braddock Road 495	Express End (near MD)	od_1025	495
2	190NO	181ND	Braddock Road 495	Express End (near MD)	od_1026	495
3	190NO	181ND	Braddock Road 495	Express End (near MD)	od_1027	495
4	190NO	181ND	Braddock Road 495	Express End (near MD)	od_1028	495

	direction	status	price	Date	ratetype
0	N	open	0.85	2021-09-26 11:30:36	DTA
1	N	open	0.90	2021-09-26 11:29:21	DTA
2	N	open	1.10	2021-09-26 11:30:36	DTA
3	N	open	1.30	2021-09-26 11:29:21	DTA
4	N	open	1.50	2021-09-26 11:29:21	DTA

Now our dataset is ready for analysis.

Reach me:

1. GitHub
2. Kaggle
3. LinkedIn

[ ]: