Build Linkedin Dataset Research

Code Explanation :

Pyhton lib selenium - to access web driver
Python Library beautifulsoup4 - used to pull the data from the HTML and XML Files

In [ ]:
```
pip install selenium
```

In [ ]:
```
pip install beautifulsoup4
```

In [ ]:
```
import os,random,sys,time
from selenium import webdriver
from bs4 import BeautifulSoup
```

Code Explanation :

Chrome driver to open the chrome window to control activities through the program
driver is the name of the folder and chromedriver is the exe file downloaded as per the
google chrome installed in my system.

In [ ]:
```
browser = webdriver.Chrome('driver/chromedriver.exe')
```

Code Explanation :
Linked In Website Link to access the site by the crawler

In [ ]:
```
browser.get('https://www.linkedin.com/login?fromSignIn=true&trk=guest_homepage-basic_nav-header-signin')
```

Code Explanation:

open command to pull the config file which has username and password

The linkedin profile will be logged in by using the email and password of t
his file

In [ ]:
```
file=open('document/config.txt')
lines=file.readlines()
username=lines[0]
password=lines[1]
```

Code Explanation:

browser.find_element_by_id() :It will let the browser get the username,passwor
d from the file previously accessed

In [ ]:
```
elementId=browser.find_element_by_id('username')
elementId.send_keys(username)
```

In [ ]:
```
elementId=browser.find_element_by_id('password')
elementId.send_keys(password)
```

In [ ]:
```
elementId.submit()
```

Code Explanation:

This file holds input url that is linked in url and reads it line by line

In [ ]:
```
file=open('C:/Users/namra/Documents/complete python and machine learning/Codin
g Ninjas/linked_Urls_To_Search.txt')
inputList=file.readlines()
info=[]
```

Code Explanation:

Page scroll function is to read the entire page of the linkedin profile

In [ ]:
```
def pageScrollFunction():
    SCROLL_PAUSE_TIME=5
    last_height=browser.execute_script("return document.body.scrollHeight")
    for i in range(3):
        browser.execute_script("window.scrollTo(0,document.body.scrollHeigh
t);")
        time.sleep(SCROLL_PAUSE_TIME)
        new_height = browser.execute_script("return document.body.scrollHeigh
t")
        if new_height == last_height:
            break
        last_height = new_height
        return 0
```

Code Explanation:

This function is called when the crawler runs to access the linkedin link and r
eads the
source code and returns it to the calling function

In [ ]:
```
def pageCallFunction(soup):
    fullLink=soup
    browser.get(fullLink)
    src = browser.page_source
    soup=BeautifulSoup(src,'lxml')
    return soup
```

Code Evaluation:

This function returns the name ,connections available on the profile

In [ ]:
```
def nameProfileConnectionFunction(soup):
    name_div = soup.find('div',{'class': 'flex-1 mr5'})
    name_loc = name_div.find_all('ul')
    name = name_loc[0].find('li').get_text().strip()
    connection=name_loc[1].find_all('li')
    connection=connection[1].get_text().strip()
    info.append(name)
    info.append(connection)
    return info
```

Code Evaluation:
Experience Block of Linked In
Position company and year of the employee in an organizaton has these commo
n code of lines so,
this function will be called when required which has reduced redundancy of t
he code

In [ ]:
```
def positionEmployerYear(soup):
    exp_section=soup.find('section',{'id' : 'experience-section'})
    exp_section=exp_section.find('ul')
    li_tags=exp_section.find('div')
    a_tags = li_tags.find('a')
    return a_tags
```

Code Evaluation:
Calls the previous fucntion and return the position of the candidate

In [ ]:
```
def positionFunction(soup):
    a_tags = positionEmployerYear(soup)
    Position = a_tags.find('h3').get_text().strip()
    info.append(Position)
    return info
```

Code Evaluation:
Company fucntion will give the details of the company of employer

In [ ]:
```
def currentEmployerFunction(soup):
    a_tags=positionEmployerYear(soup)
    Current_Employer=a_tags.find_all('p')[1].get_text().strip()
    info.append(Current_Employer)
    return info
```

Code Evaluation:
Range of Dates of service in the organization

In [ ]:
```
def departmentDatesFunction(soup):
    a_tags=positionEmployerYear(soup)
    departmentDates=a_tags.find_all('h4')[0].find_all('span')[1].get_text().st
rip()
    info.append(departmentDates)
    return info
```

Code Evaluation:

This functin will provide education background information for the candidat
e.

college name

degree name

Range of year of education

In [ ]:
```
def educationFunction(soup):
    edu_section=soup.find('section',{'id' : 'education-section'}).find('ul')
    college = edu_section.find('h3').get_text().strip()
    degree_name = edu_section.find('p',{'class':'pv-entity__secondary-title pv
-entity__fos t-14 t-black t-normal'}).find_all('span')[1].get_text()
    degree_year = edu_section.find('p',{'class':'pv-entity__dates t-14 t-black
--light t-normal'}).find_all('span')[1].get_text().strip()
    info.append(college)
    info.append(degree_name)
    info.append(degree_year)
    return info
```

```
                Code Evaluation:
       This code will be executed by calling the previous defined functions and wil
l display the return
       values of the functions.
       inputlist = The list of the array execution.
       info will dispaly the entire fetched output

      pageCallFunction -> will call the linkedin input url on as per the current lo
op value and receive
                           the page source code in the soup

       nameProfileConnectionFunction->It will fetch the name and the total number o
f connections on the
       current profile return the name and connections in the soup

       positionFunction -> It will return the position or designation of the candia
dte in the soup

       currentEmployerFunction ->The workplace of the candidate will be received in
the soup

       departmentDatesFunction -> The tenure of date will be retreived

       educational information  -> Includes the candidates college name,degree name
and range of degree date

       educationFunction -> This will display the collaboration of previous and edu
cational
                              information of the candidate  by appending it into info
which is
                              the return value of previous functions
```

In [ ]:
```
for i in inputList:
    soup=pageCallFunction(i)
    pageScrollFunction()
    nameProfileConnectionFunction(soup)
    positionFunction(soup)
    currentEmployerFunction(soup)
    departmentDatesFunction(soup)
    info=educationFunction(soup)
print(info)
print()
```

Code Evaluation:
This code returns the array in the tsv format
tsv.shape -> returns the total number of elements
tsv.reshape -> is used to restructure the data as per our requirements b
y provinding row,column data

In [ ]:
```
import numpy as np
a=[]
tsv = np.append(a,info)
tsv
tsv.shape
tsv.reshape(800,9)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: