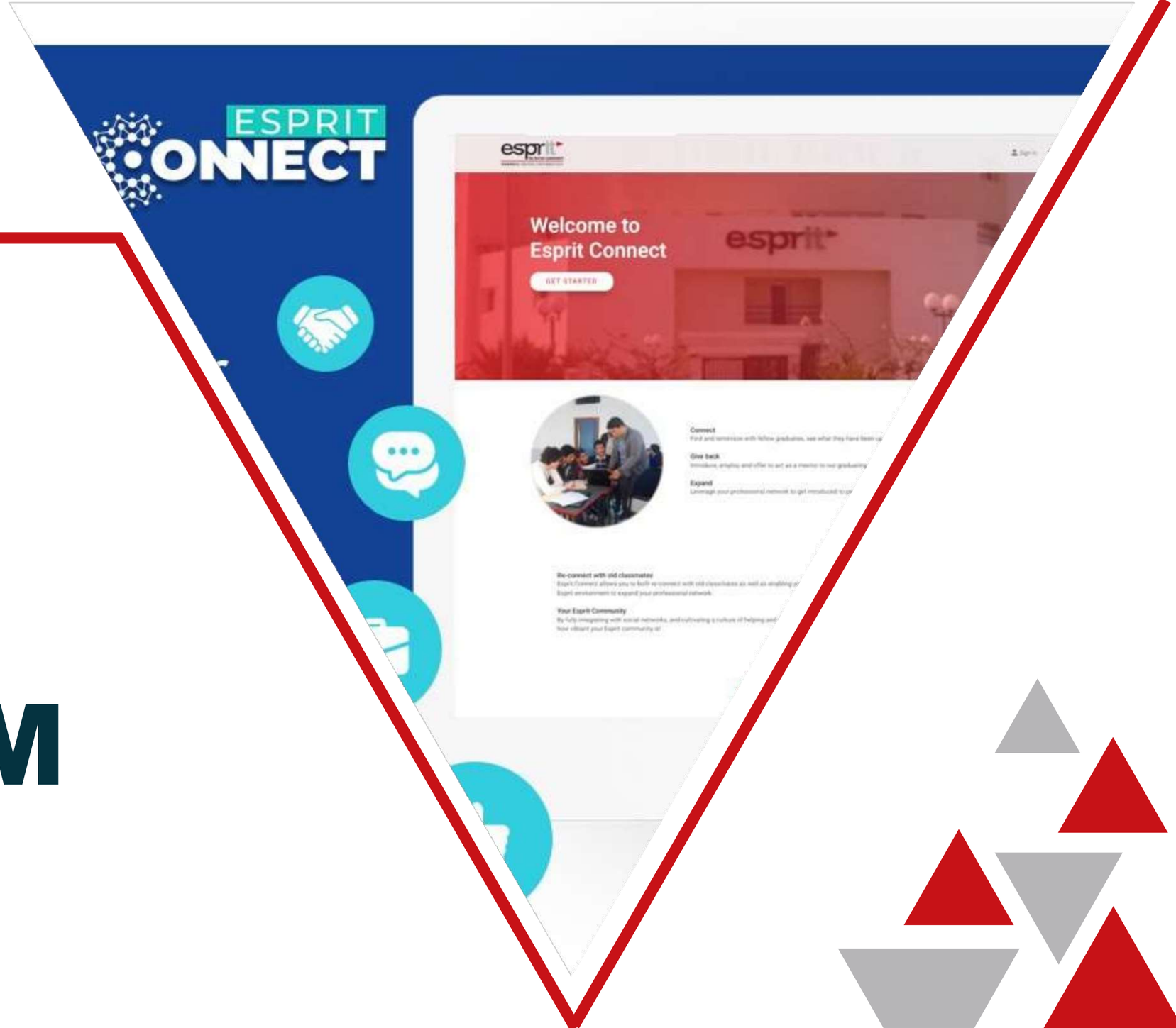
esprit
Se former autrement
HONORIS UNITED UNIVERSITIES

CAREER CENTER PLATEFORM



PLAN

01 IBM Master Plan

02 Functional Requirments

03 Data Understanding

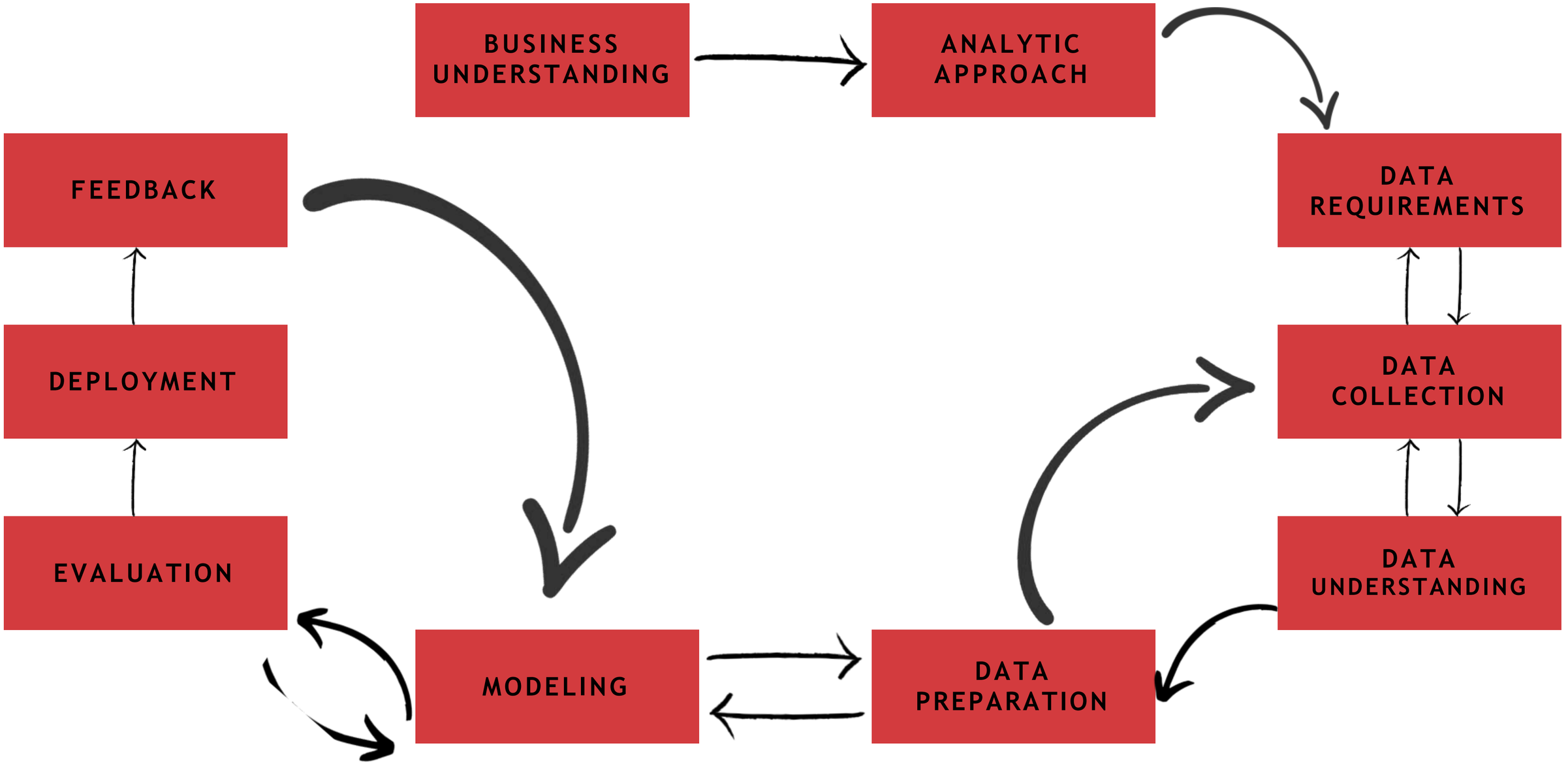
04 Data Preparation

05 Modelling

06 Deployment

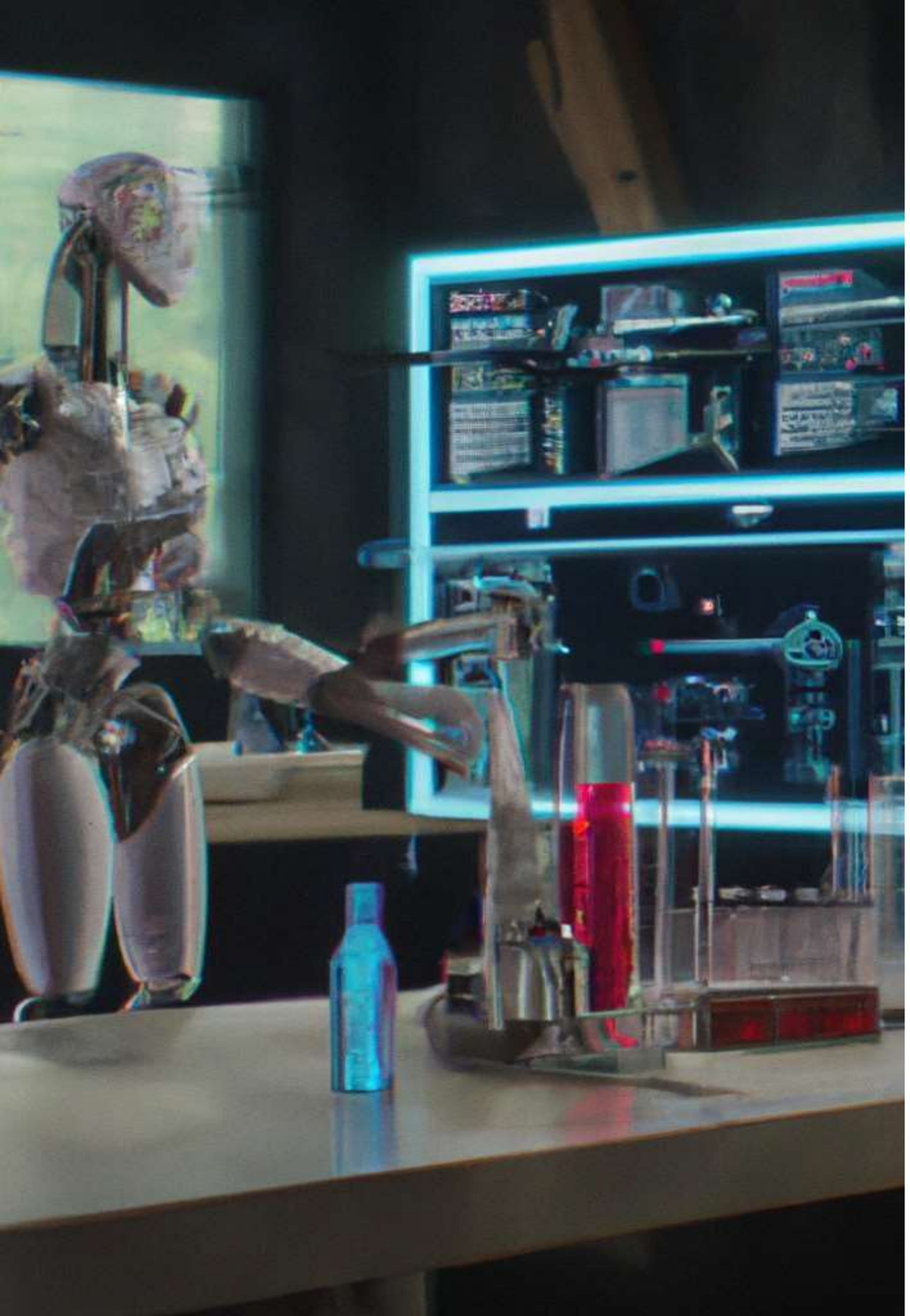
IBM MASTER PLAN

01


BUSINESS UNDERSTANDING
ANALYTIC APPROACH
DATA REQUIREMENTS
DATA COLLECTION
DATA UNDERSTANDING
DATA PREPARATION
MODELING
EVALUATION
DEPLOYMENT
FEEDBACK

FUNCTIONAL REQUIRMENTS

02



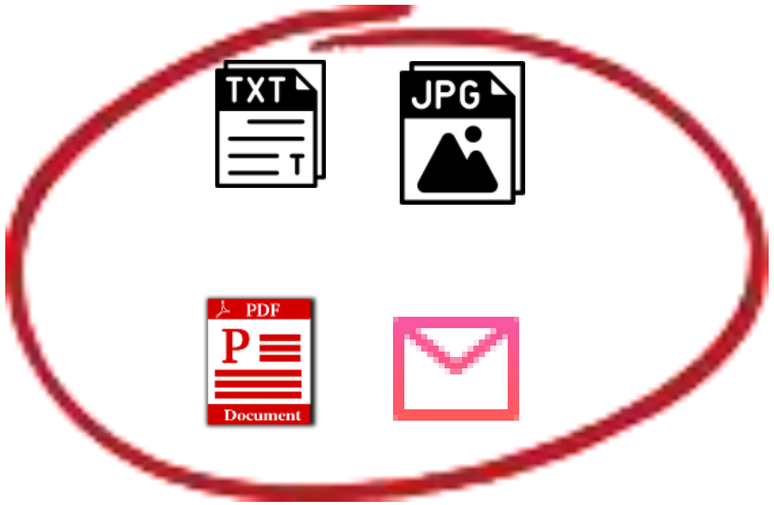
Functional Requirments :

01 Business Objectives

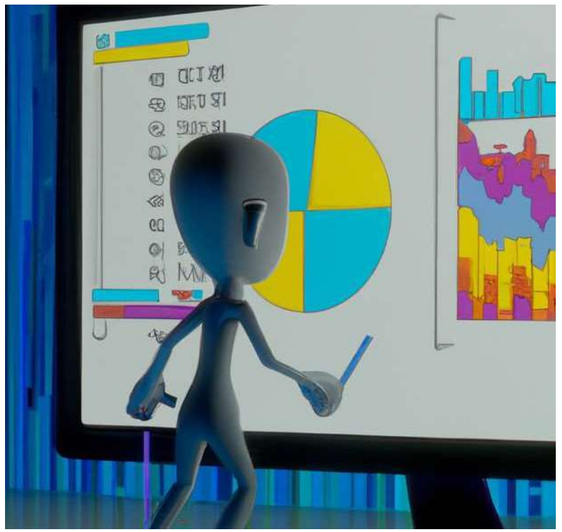
02 Data science objectives

PROBLEMS

Difficulty in data management



Receives job offers in various formats



Lack of data analytics capabilities

Ineffective matching of candidates and job offers

TARGETED CUSTOMERS

	Student at Esprit (TIC, EM ,CG)
	ESPRIT Alumni
	Employability Pole of the Esprit Group

01 Business Objectives :

Profilifig

A solution that can efficiently and effectively manage the large volume of job and internship offers received in different formats.

M6tchifig 6fid recomm6fid6tiofi cofidid6tes

A system that can match candidates with suitable job offers, with a certain percentage based on their skills and qualifications.

→ Reduce the amount of time required to find suitable job opportunities/profile

01 Business Objectives :

Reportifig 6fid predectifig

A solution that provides real-time data analytics and reporting capabilities based on KPI's :
-Number of offers per period (month, quarter, year)
-number of offers by Technologies,trades or fields
-the geographical distribution of the job offers nationally and internationally

→ Give the EP more visibility into the job maket

02 Data science objectives :

Descriptive 6fi6lytics

-Web scraping can be used to collect cv's from Linkedin.
-OCR and NLP extarct and summarize informations from PDF documents..
-data visualisation can be used to understand the geographical distribution of the job offers.

Predictive 6fi6lytics

-machine learning algorithms can be used to find the best candidates for a job offer.
-NLP can be used to match job offers with CVs

02

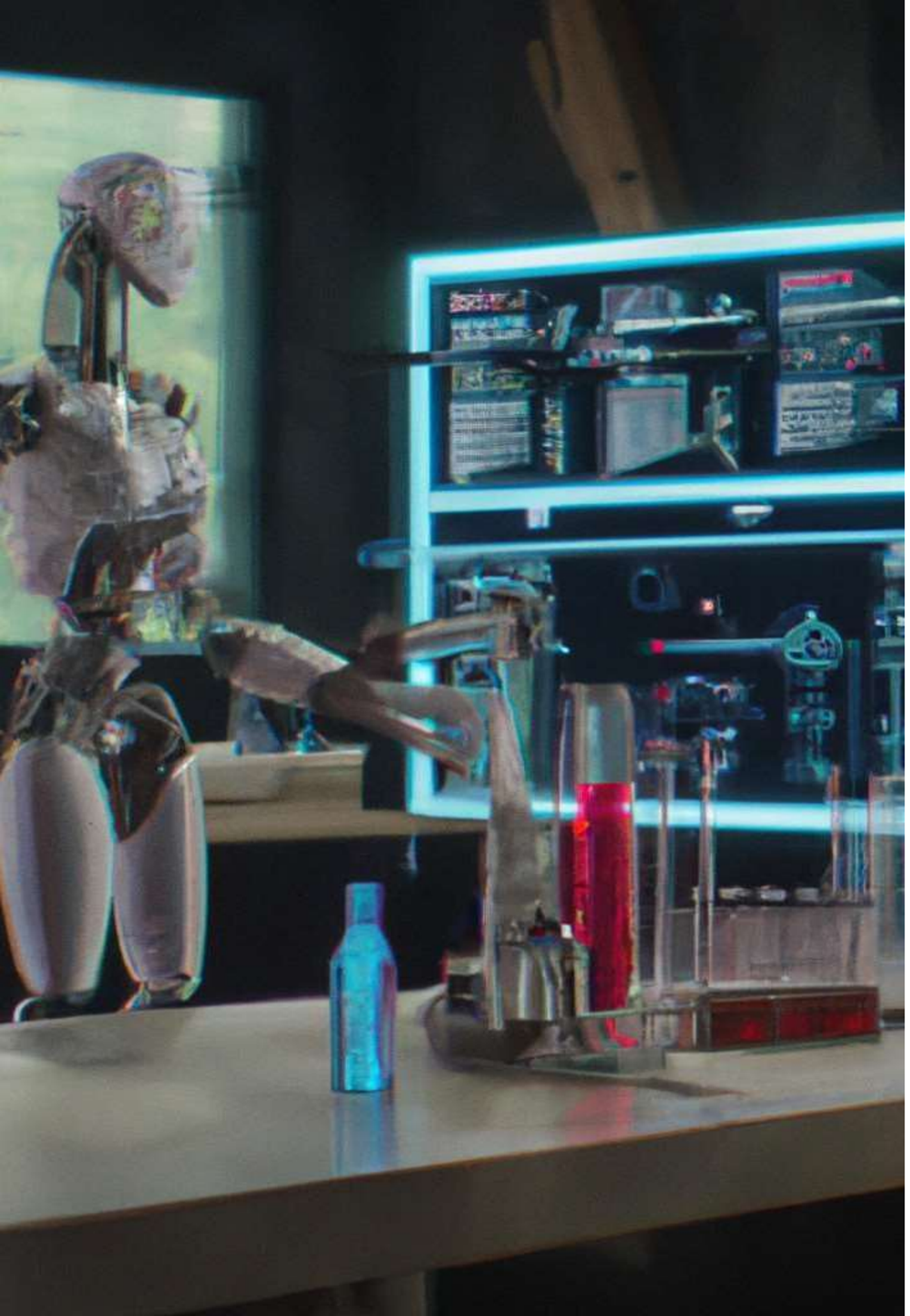
Data science objectives :

D6t6 extr6ctiofi

Extracting insights and trends from the data, such as the skills required by the job market and the geographical distribution of the offers.

DATA UNDERSTANDING

03

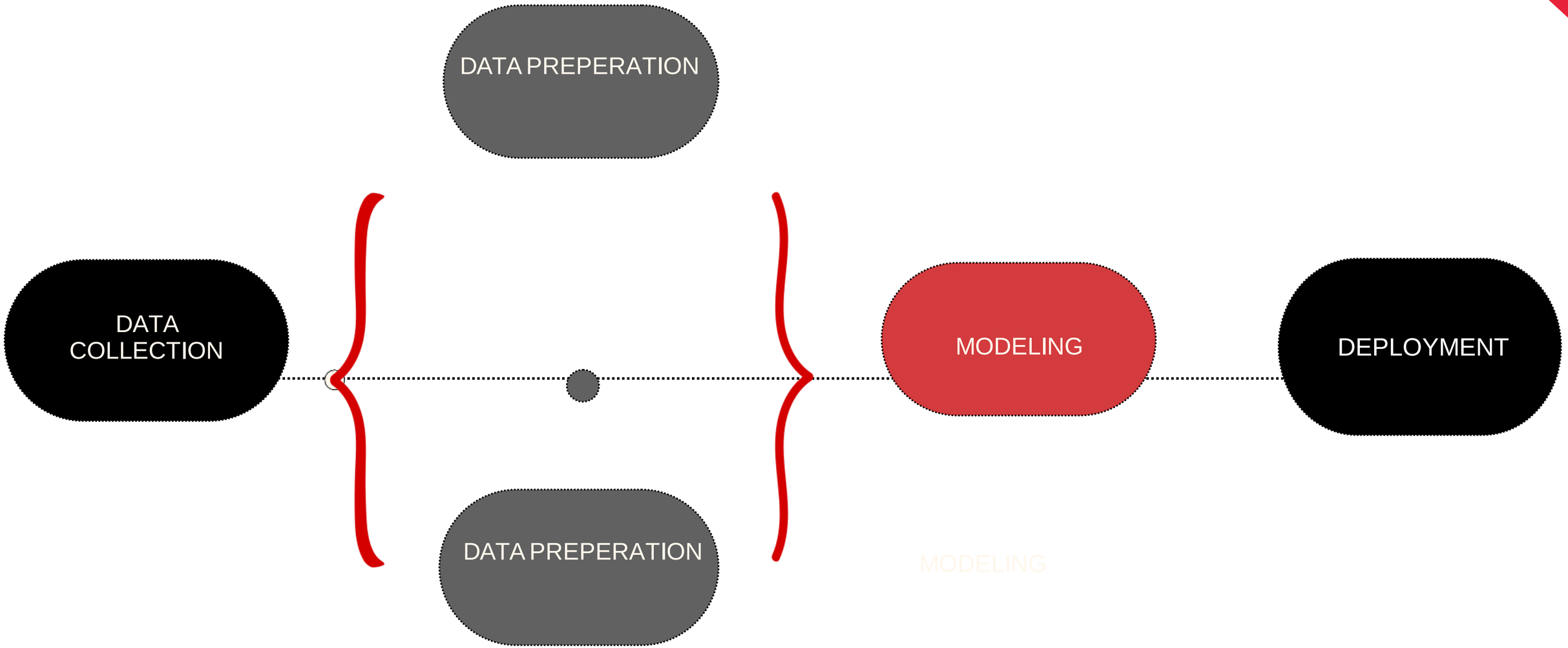


Data Understanding :

01 Data Sources

02 Data Quality

Students Data



Jobs data

Data Sources :

01 students data from linkedin

02 job data from emails

03 job data from esprit connect

04 job data from PDF

STUDENTS DATA FROM LINKEDIN

```
        # xpath to extract the text from the class containing the name
        name = sel.xpath('//*[starts-with(@class, "text-heading-xlarge inline t-24 v-align-middle break-word")]/text()').ext

        # if name exists, strip it of any leading/trailing spaces and add it to the linkedin_data list
        if name:
            name = name.strip()


except Exception as e:
    # handle any exceptions that occur during scraping
    print(f"Error scraping {url}: {e}")


# xpath to extract the text from the class containing the job title
job_title = sel.xpath('//*[starts-with(@class, "text-body-medium break-words")]/text()').extract_first()

if job_title:
    job_title = job_title.strip()


try:
    # xpath to extract the text from the class containing the company name
    company = driver.find_element(By.XPATH,'//ul[@class="pv-text-details__right-panel"]').text

except:
    company = 'None'

    if company:
        company = company.strip()


# xpath to extract the text from the class containing the location
location = sel.xpath('//*[starts-with(@class, "text-body-small inline t-black--light break-word")]/text()').extract()

if location:
    location = ", ".join(location).strip()
```

STUDENTS DATA FROM LINKEDIN

	Unnamed: 0	Name	job_title
0	0	tlili khaled	Full Stack Web Developer
1	1	Khalil Ben said	Web developer at ITSolution
2	2	Iheb Mejri	Full Stack Developer
3	3	Omar Talbi	Freelance Web Developer
4	4	Nahla SHIRI	Frontend Web Developer
5	5	Safwen Dammak	Freelance Web Developer
6	6	Mehdi Othman	Web Developer
7	7	omar mhiri	Web and mobile developer
8	8	Mohamed Bechir Lahmer	Web developer at Mobelite.
9	9	No results	No results

	company	location
0	Sofrecom Tunisie\nInstitut supérieur d'informa...	Gouvernorat Tunis, Tunisie
1	ITSolution Tunisia\nEcole Supérieure Privée d'...	Gouvernorat Ben Arous, Tunisie
2	Tutosh	Délégation Bizerte Sud, Gouvernorat Bizerte, T...
3	North American Private University: Internation...	Gouvernorat Sfax, Tunisie
4	l'Ecole Supérieure de Technologie et d'Informa...	Gouvernorat Ben Arous, Tunisie
5	Université Sesame	Gouvernorat Sfax, Tunisie
6	OpenClassrooms	Gouvernorat Tunis, Tunisie
7	Freelance | Self-Employed	Gouvernorat Tunis, Tunisie
8	Mobelite Tunisie	Gouvernorat Tunis, Tunisie
9	None	No results

	URL:	skills:
0	https://tn.linkedin.com/in/khaled-tlili-85b887...	PHP, Android Development, Mobile Applications,...
1	https://tn.linkedin.com/in/khalil-ben-said-487...	Adobe Photoshop, Problem Solving, English, Des...
2	https://tn.linkedin.com/in/iheb-mejri/en	NestJS, GraphQL, PostgreSQL, RxJS, Prototyping...
3	https://tn.linkedin.com/in/omar-talbi-sfax	Laravel, React, JavaScript, PHP, Microsoft Bot...
4	https://tn.linkedin.com/in/nahla-shiri	MongoDB, HTML5, AngularJS, JavaScript, jQuery,...
5	https://tn.linkedin.com/in/safwendammak	Angular, Applications web, Conception techniqu...
6	https://tn.linkedin.com/in/othmanmahdi/en?trk=...	HTML5, CSS3, PHP, Web Services, Web Applicatio...
7	https://tn.linkedin.com/in/omar-mhiri	HTML5, PHP, Shell Scripting, Cloud Computing, ...
8	https://tn.linkedin.com/in/mohamed-bechir-lahm...	Laravel, Symfony, Web Development, MySQL, PHP,...
9	https://tn.linkedin.com/in/labib-yanes-18b528203	PrestaShop, WordPress, SQL, Git, MongoDB, Web ...

	experiences:
0	Stagiaire
1	Web Developer, Trainee, Trainee
2	NaN
3	Freelance Web Developer, Web Development Inter...
4	Frontend Web Developer, Full Stack Developer, ...
5	Web Developer, Web Development Intern, Web Dev...
6	Magasinier
7	Web Designer, Web and mobile developer, Web an...
8	Web Developer, FullStack web developer, FullSt...
9	Web Developer, Web Developer, Web Integrator

STUDENTS DATA FROM LINKEDIN

→ **It shows :**

- **N6me:** The name of the candidate,it helps identify the individual and personalize any communication
- **Job title:** The job title or current position of the candidate
- **comp6fiy:** The name of the company or school the candidate has worked for or attended
- **Loc6tiofi:** The candidate's location
- **URL:** The profile URL is a unique identifier that links directly to the candidate's online profile.
- **Skills:** The skills of the candidate
- **Experiefices:** career history, level of seniority, industry experience, and areas of expertise.

JOB DATA FROM EMAILS :

1-the PE receives all the offers from different recources by mail

2-Collecting those emails that were received and broadcasted to the students.

we import the imaplib library, which allows us to connect to the Gmail server and retrieve emails.

- we login to a gmail server with one of our users
- Search for emails with specific key and value
- we displayed the fetched messages recieved by
- <pole-employabilite-esprit@esprit.tn> in that inbox.

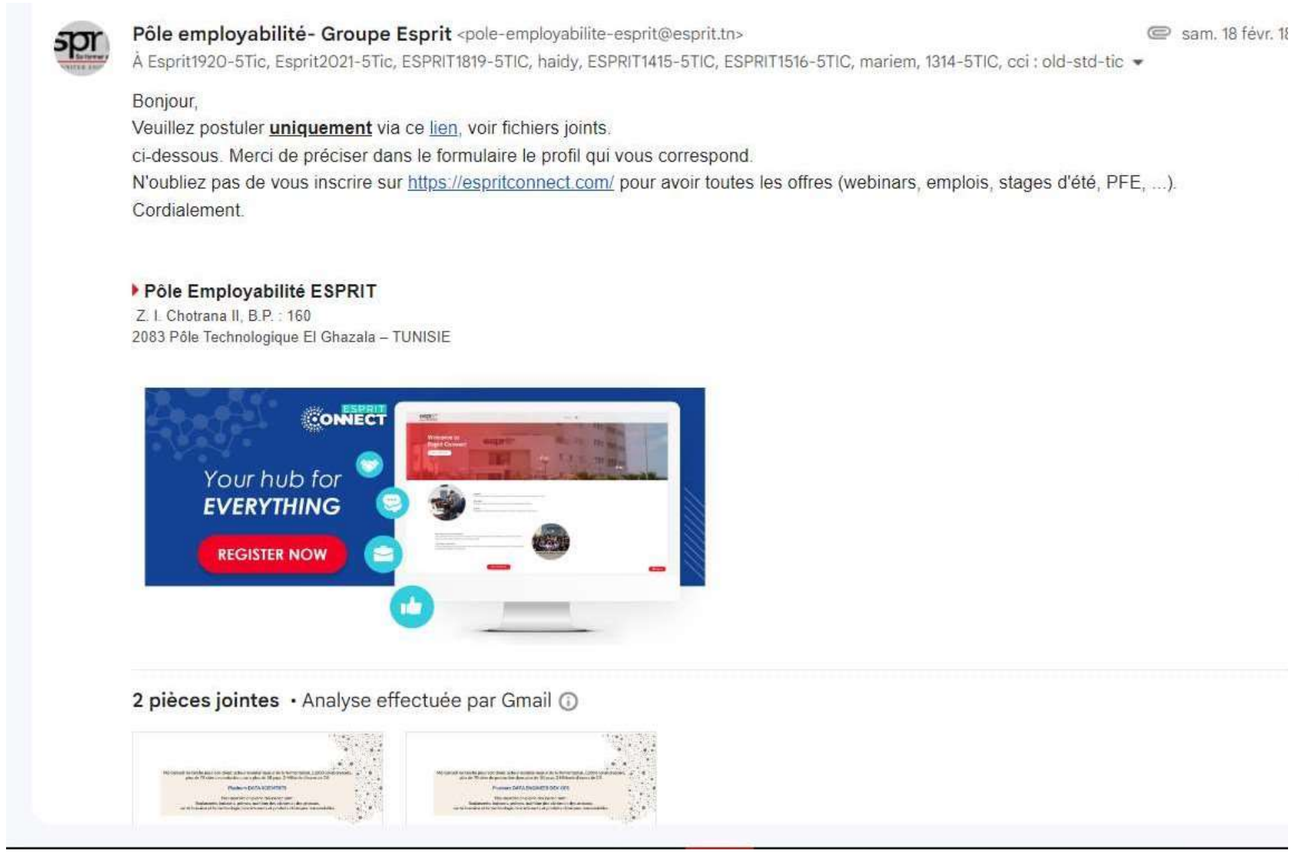
```
[ ]  user = 'eya.guirat@esprit.tn'
     password = 'ksos bcva vfub yfdv'

[ ]  imap_url = 'imap.gmail.com'

[ ]  my_mail = imaplib.IMAP4_SSL(imap_url)

     gmail = imaplib.IMAP4_SSL("imap.gmail.com", '993')
     gmail.login(user,password)
     gmail.select("")

     #head, data = gmail.search(None, 'ALL')

[ ]  # Log in using your credentials
     my_mail.login(user, password)

     ('OK', [b'eya.guirat@esprit.tn authenticated (Success)'])

[ ]  # Select the Inbox to fetch messages
     my_mail.select('Inbox')

     ('OK', [b'5035'])

     #Define Key and Value for email search
     #For other keys (criteria): https://gist.github.com/martinrusev/6121028#file-imap-search
     key = 'FROM'
     value = 'pole-employabilite-esprit@esprit.tn'
     _, data = my_mail.search(None, key, value)  #Search for emails with specific key and value

     mail_id_list = data[0].split()  #IDs of all emails that we want to fetch
```

After fetching all the messages: we've notcied that they contain a lot of unwanted details in text form

-> extracting the wanted elements

- **subject**
- **from**
- **body**

1 text
2 pdf
3 images



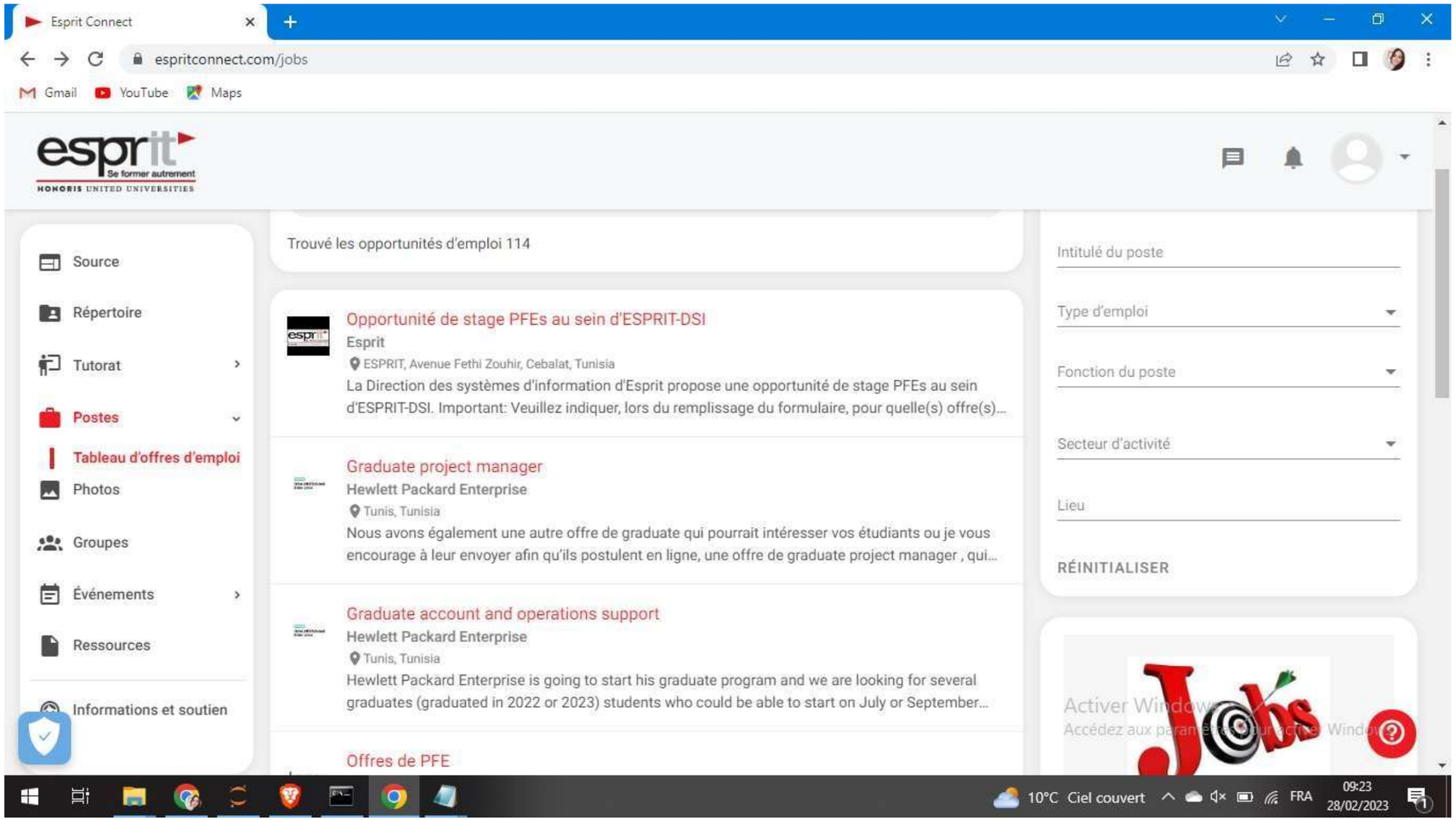
```
my_msg=email.message_from_bytes((response_part[1]))
print("_____")
print ("subj:", my_msg['subject'])
msgsubj.append(my_msg['subject'])
print ("from:", my_msg['from'])
msgfrom.append(my_msg['from'])
strr=""
t+=1
k=0
print ("body:")
for part in my_msg.walk():
    print(part.get_content_type())
    if part.get_content_type() == '\n' or part.get_content_type() == '\n':
        strr+="  "
    strr+=(part.get_content_type())
    if part.get_content_type() == 'text/plain':
        print (part.get_payload())
        strr+=(part.get_payload())
    if part.get_content_type() == 'application/pdf' or part.get_content_type() == 'image/*':
        pdf.append(t)
        payload = part.get_payload(decode=True)
        filename = part.get_filename()
        if payload and filename:
            # Sanitize the filename
            filename = re.sub(r'[^0-9a-zA-Z]+', '', filename) + ".pdf"
            with open(filename, 'wb') as f:
                try:
                    f.write(payload)
                except Exception as e:
```

```
_____
subj: Offres d'emploi-Banque de Tunisie
from: =?UTF-8?Q?P=C3=B4le_employabilit=C3=A9=2D_Groupe_Esprit?= <pole-employabilite-esprit@esprit.tn>
body:
multipart/alternative
text/plain
Bonjour,
Veuillez postuler *uniquement* via ce lien
<https://docs.google.com/forms/d/e/1FAIpQLSeHjB2du8pwq3xQJLTHWrL8BBH5EWnc0S=
25sYjURrAkHKURfQ/viewform>,
voir ci-dessous.
N'oubliez pas de vous inscrire sur https://espritconnect.com/ pour avoir
toutes les offres (webinars, emplois, stages d'=C3=A9t=C3=A9, PFE, ...).
Cordialement.

**********_
```

Out[652]:

	type	from	sujet	skills	link
0	job	<pole-employabilite-esprit@esprit.tn>	Offres d'emploi-Banque de Tunisie	[net, core, entity, framework, vmc, javascript]	[https://docsgooglecom/forms/d/e/, https://esp...
1	Unknown	<pole-employabilite-esprit@esprit.tn>	erreur	[]	[https://wwwcofomocom/f, https://docsgooglecom...
2	job	<pole-employabilite-esprit@esprit.tn>	Offres d'emploi-Groupe Lesaffre-Data Scientist...	[htmlapplication, pdfapplication, pdf]	[https://docsgooglecom/forms/d/e/, https://esp...
3	job	<pole-employabilite-esprit@esprit.tn>	Offres d'emploi-ODDO BHF	[recruited, developers, dotnet, engineers, dat...	[https://docsgooglecom/forms/d/e/, https://esp...
4	Unknown	<pole-employabilite-esprit@esprit.tn>	Hewlett Packard Enterprise recruits a graduate...	[plainfor, encourage, line, manager, basically...	[https://espritconnectcom/, https://careershpe...
...	...	...	...	...	...
147	Unknown	<pole-employabilite-esprit@esprit.tn>	erreur	[plainfor, forwarded, message, bousbia, mar, f...	[https://formsgle/, https://formsgle/, https:/...
148	Intership	<pole-employabilite-esprit@esprit.tn>	Virtual Internships	[minute, ce, formu, lar, jfe, collaboration]	[https:///jfe/form/, https://wwwvirtualinterns...

JOB DATA FROM ESPRIT CONNECT :



Entrée [3]:

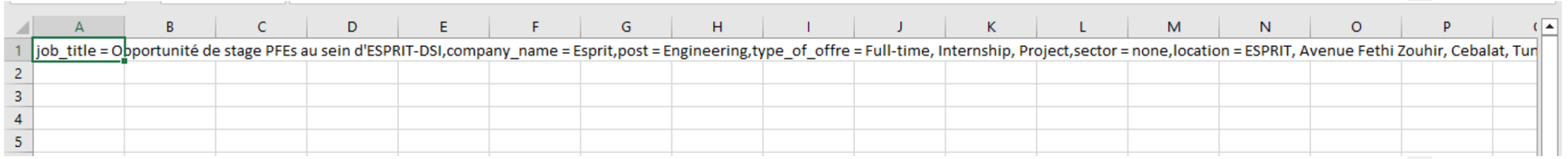
```
def findjob():
    opts = Options()
    opts.add_experimental_option("debuggerAddress","localhost:9250")
    driver = webdriver.Chrome(options=opts, executable_path='chromedriver')

    driver.get('https://espritconnect.com/jobs')

    for i in range(397, 272, -1):
        driver.get(f'https://espritconnect.com/jobs/{i}')
        time.sleep(random.uniform(2.5, 4.9))

        soup = BeautifulSoup(driver.page_source, 'lxml')
        data = {}

        try:
            job_title = soup.find('h2', {'id': 'jobPageJobTitle'}).text.strip()
        except:
            job_title = 'none'

        try:
            company_name = soup.find('p', {'id': 'jobPageOrganization_0'}).text.strip()
        except:
            company_name = 'none'

        try:
            post = soup.find('p', {'id': 'jobPageJobFunction_0'}).text.strip()
        except:
            post = 'none'
```

```
        try:
            sector = soup.find('p', {'id': 'jobPageJobFunction_2'}).text.strip()
        except:
            sector = 'none'

        try:
            location = soup.find('span', {'class': 'mat-caption location-address gw-vertical-align-middle location-icon-text'}).t
        except:
            location = 'none'

        try:
            url = soup.find('a', {'id': 'JobPageJobDescriptionFilePath'})['href']
        except:
            url = 'none'

        filename = f'c:/posts/{i}.txt'
        with open(filename, 'w') as f:
            f.write(f"job_title = {job_title}\n")
            f.write(f"company_name = {company_name}\n")
            f.write(f"post = {post}\n")
            f.write(f"type_of_offre = {type_offre}\n")
            f.write(f"sector = {sector}\n")
            f.write(f"location = {location}\n")
            f.write(f"url = {url}\n")

        print(f'File saved: {filename}')

driver.quit()
```

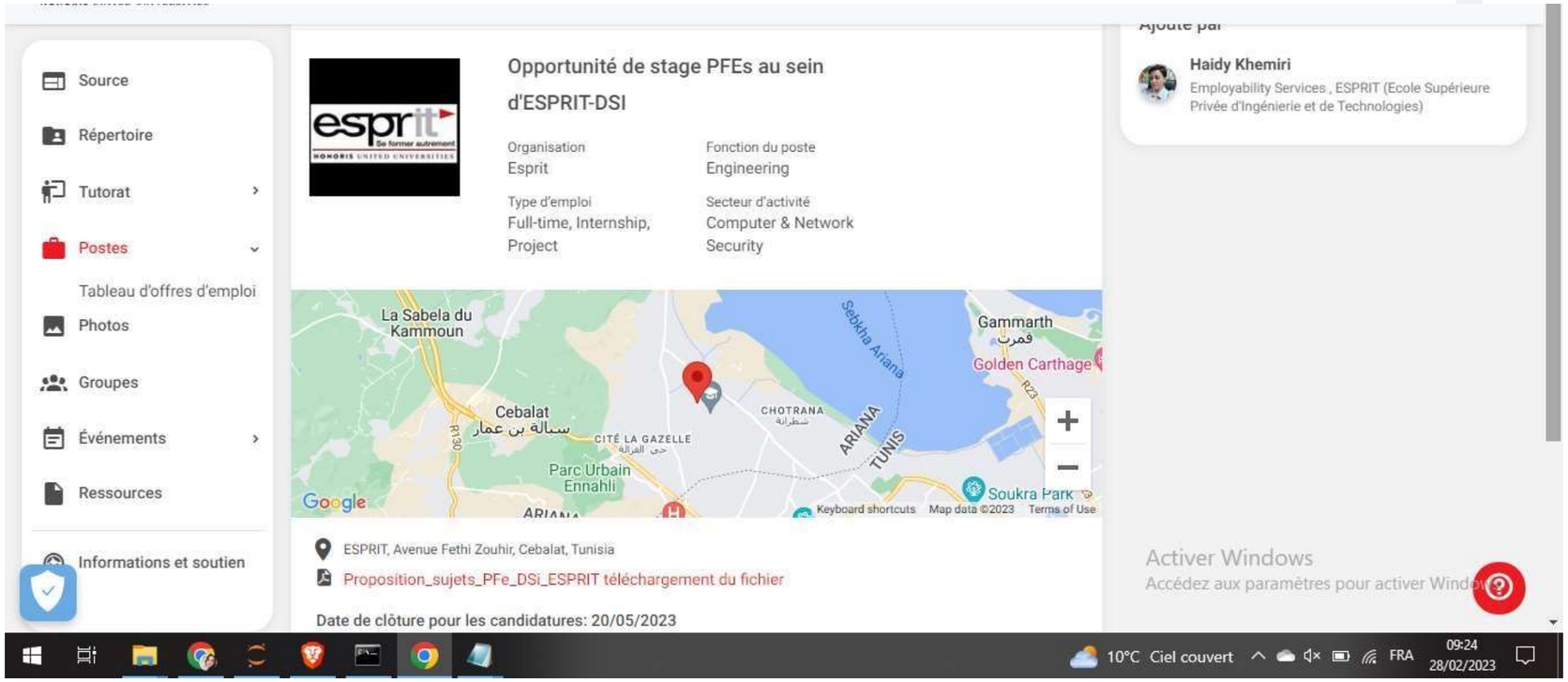
397.txt - Bloc-notes

Fichier Edition Format Affichage Aide

```
job_title = Opportunité de stage PFEs au sein d'ESPRIT-DSI
company_name = Esprit
post = Engineering
type_of_offre = Full-time, Internship, Project
sector = Computer & Network Security
location = ESPRIT, Avenue Fethi Zouhir, Cebalat, Tunisia
url = https://dx5i3n065oxey.cloudfront.net/platform/50238/job/original/c798a963-d9c8-4b87
```



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	job_title = Opportunité de stage PFEs au sein d'ESPRIT-DSI,company_name = Esprit,post = Engineering,type_of_offre = Full-time, Internship, Project,sector = none,location = ESPRIT, Avenue Fethi Zouhir, Cebalat, Tun															
2																
3																
4																
5																



JOB DATA FROM PDF :

```python
def convert_pdf(path):
    l=path.split('\\')
    file=l[-1]
    pdf_name=file.split('.')[0]
    #print(pdf_name)
    images=convert_from_path(path)

    save_path="Data\\Converted_images\\"
    n=0
    for i,image in enumerate(images):
        fname = "image_"+pdf_name+"_"+ str(i) + ".png"
        image.save(save_path+fname, "PNG")
        n=n+1
    return n

def is_scanned_pdf(file_path):
    """
    Determines if a PDF file is scanned or not by checking if it contains text or not.

    Parameters:
        file_path (str): The path to the PDF file to check.

    Returns:
        bool: True if the PDF file is scanned, False otherwise.
    """
    with fitz.open(file_path) as pdf:
        for page in pdf:
            text = page.get_text()
            if text.strip() != "":
                # If the page contains text, it's not a scanned PDF
                return False
    # If all pages don't contain text, it's a scanned PDF
    return True
```

JOB DATA FROM PDF :

```python
import PyPDF2
#from googletrans import Translator
import numpy as np
import cv2
from PIL import Image
import pytesseract

pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'


def ocr_core(image):
    text = pytesseract.image_to_string(image, lang='fra', config='-c page_separator=""')
    return text


def get_grayscale(image):
    return cv2.cvtColor(image, cv2.COLOR_BGR2HSV)


def remove_noise(image):
    return cv2.medianBlur(image, 5)


def thresholding(image):
    # need thrershold optimization !!!!!!!@TODO/TO SEARCH
    return cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)


def ocr_extract_txt(image_name):
    img = cv2.imread('Data\\Converted_images\\' + image_name)
    img = cv2.resize(img, (0, 0), fx=0.9, fy=0.9)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    text_file = open("Data\\Converted_images\\AllText.txt", "w")
    text_file.write(ocr_core(img[1]))
    return ocr_core(img)
```

JOB DATA FROM PDF :

```python
def extract_text_from_pdf(pdf_file_path):
    text = ""

    with open(pdf_file_path, 'rb') as pdf_file:
        pdf_reader = PyPDF2.PdfReader(pdf_file)

        # Iterate through each page in the PDF file
        for page_num in range(len(pdf_reader.pages)):
            page = pdf_reader.pages[page_num]

            # Extract text from the page
            page_text = page.extract_text()

            # Append the page text to the overall text string
            text += page_text

    return text


def extract(pdf_file_path):
    if is_scanned_pdf(pdf_file_path):
        convert_pdf(pdf_file_path)
        l = pdf_file_path.split('\\')
        file = l[-1]
        pdf_name = file.split('.')[0]
        t=""
        for i in range(convert_pdf(pdf_file_path)):
            fname = "image_" + pdf_name + "_" + str(i) + ".png"
            #print(fname)
            t=t+ocr_extract_txt(fname)
        return t


    else :
        return extract_text_from_pdf(pdf_file_path)
```

```python
list_of_text=[]
paths_corrected=[]
for i in range(len(paths)):
    if(paths[i].split('.')[-1]=='pdf'):
        print(paths[i])
        paths_corrected.append(paths[i])
        lop=extract('data\\convertedPDF\\'+paths[i]).split('\n')
        list_of_text.append(lop)
```

JOB DATA FROM PDF :

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
#nltk.download('stopwords')
import re
```

```
#print(filtered_lop)
def st(a):
    return a.strip()
cleaned_list=[]
for i in range(len(list_of_text)):
    filtered_lop=[string for string in list_of_text[i] if string.strip() and not string.strip().isdigit()]
    cleaned_lop=list(map(st,filtered_lop))
    raw_txt="".join(cleaned_lop)
    text_tokens = word_tokenize(raw_txt)
    tokens_without_sw = [word for word in text_tokens if not word in stopwords.words()]
    txt="".join(tokens_without_sw)
    cleaned_list.append(txt)
```

```
print(cleaned_list[2])
```

```
SANTANDERTECHHUB-PROFILESProfiledescription.1.DevSecOps.WHATYOUWILLBEDOING.AsDevSecOpsEngineermissioncollaborate-to-endDevOpspr
ojects,contributingfullviewarchitecture,design,deploymentsolutions,systemintegration,automation,executionsupport.Wedifferentfro
nts:-PipelinedeploymentsALMtools:Automaticsoftwarepromotionbetweendifferentenvironments.Orchestrationqualitycontrolsystems,incl
udingcybersecurity.-SWdeployments.-Testing:Automationfunctionalregressionstsensurecorrectbehaviorapplication.-Systemintegratio
n/continuousintegration:provideteamsneedtointegratecodedailybasisunittestautomationsoftwarequality.-Environmentsinfrastructure
s:Makeavailable,maintainscalewhennecessary-configurationenvironmentsinfrastructuresthattheteamdevelopsolution.-Continuousmonito
ring.WHATWEARELOOKINGFOR.Requiredqualifications.-ASTEMbachelor/degree/master(Science,Technology,EngineeringandMathematics).Grad
uatesMathematics,ComputerEngineering,Telecommunications,Statistics,Physicssimilar.-Pasionlatesttrends,technologiesprogramminglа
nguages.-Teamspirit,communicationskills,interpersonalskills,-Problemsolver,creative,goaloriented.-Eagerlearn.-English:B2/C1.Kno
wledgesubjectsinteresteddeveloping.-Automaticdeploymenttools(Terraform,Ansible,.).-Cloudarchitectures(vCloud,Azure/AWS),Monitor
ingtools.-OpenShift/DockerHub/Kubernetes/CloudALM(AzureDevOps,DevOpsAWS,GitHub,GitLab,Atllassian,.).-Jira/Confluence,Git,Jenkin
s(CloudBees),Nexus,SonarHarbour.-Scriptinglanguages(Groovy,Python,JavaScript,.).-Databases(PostgreSQL,MongoDB,AzureCosmosDB).
```

JOB DATA FROM PDF :

```
import openai
openai.api_key = "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"
list_of_gpt=[]

for i in range(len(cleaned_list)):
    try:
        stringLen=len(cleaned_list[i])
        if stringLen%2 == 0:
             firstString = slice(0, stringLen//2)
             secondString = slice(stringLen//2, stringLen)
             first=cleaned_list[i][firstString]
             second=cleaned_list[i][secondString]
        else:
            firstString = slice(0, stringLen//2)
            secondString = slice(stringLen//2, stringLen)
            first=cleaned_list[i][firstString]
            second=cleaned_list[i][secondString]
        response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
                    {"role": "system", "content": "You are a chatbot"},
                    {"role": "user", "content": "Give me the list of job offers and their required skills in the given text trar
                    {"role": "user", "content":"your response must be in the form of a python's list of  json objects with only
                    {"role": "user", "content": "the json in the response must be will formatted wat i mean is it must have \" a
                    {"role": "user", "content": first},
                    {"role": "user", "content": second},
                ]
            )

        result = ''
        for choice in response.choices:
            result += choice.message.content
            list_of_gpt.append(result)
    except:
        print("error")
```

JOB DATA FROM PDF :

```
[
 {
  "job title": "Data Management Solution Delivery",
  "skill required": "Cloudera/Spark/SnowFlake/Databricks/Stratio/Hive/Impala/HBase, Spark Streaming/Flink/Storm, API Ecosystem
s, HDFS, S3, Kafka, Scala/Python/Java, CI/CD, Google/AWS/Azure",
  "location": "SantanderTechHub"
 },
 {
  "job title": "Data Pipeline Developer",
  "skill required": "Bachelor/Degree/Master in STEM (Science, Technology, Engineering and Mathematics), latest trends, technolo
gies and programming languages, team spirit, communication skills, interpersonal skills, problem solver, creative, goal oriente
d, eager to learn, English: B2/C1",
  "location": "SantanderTechHub"
 }
]
```

```
5]: import pandas as pd
    import json
```

```
5]: frames=[]
    for i in range(len(list_of_json)):
        try:
            link_column=[]
            nbr=list_of_json[i].count('"job title"')
            for d in range(nbr):
                link_column.append(paths_corrected[i])
            print(i)
            js=json.loads(list_of_json[i])
            df=pd.DataFrame.from_records(js)
            df.columns=['job_title','skill_required','location']
            df['link']=link_column
            frames.append(df)
        except:
            print('need formatting')
```

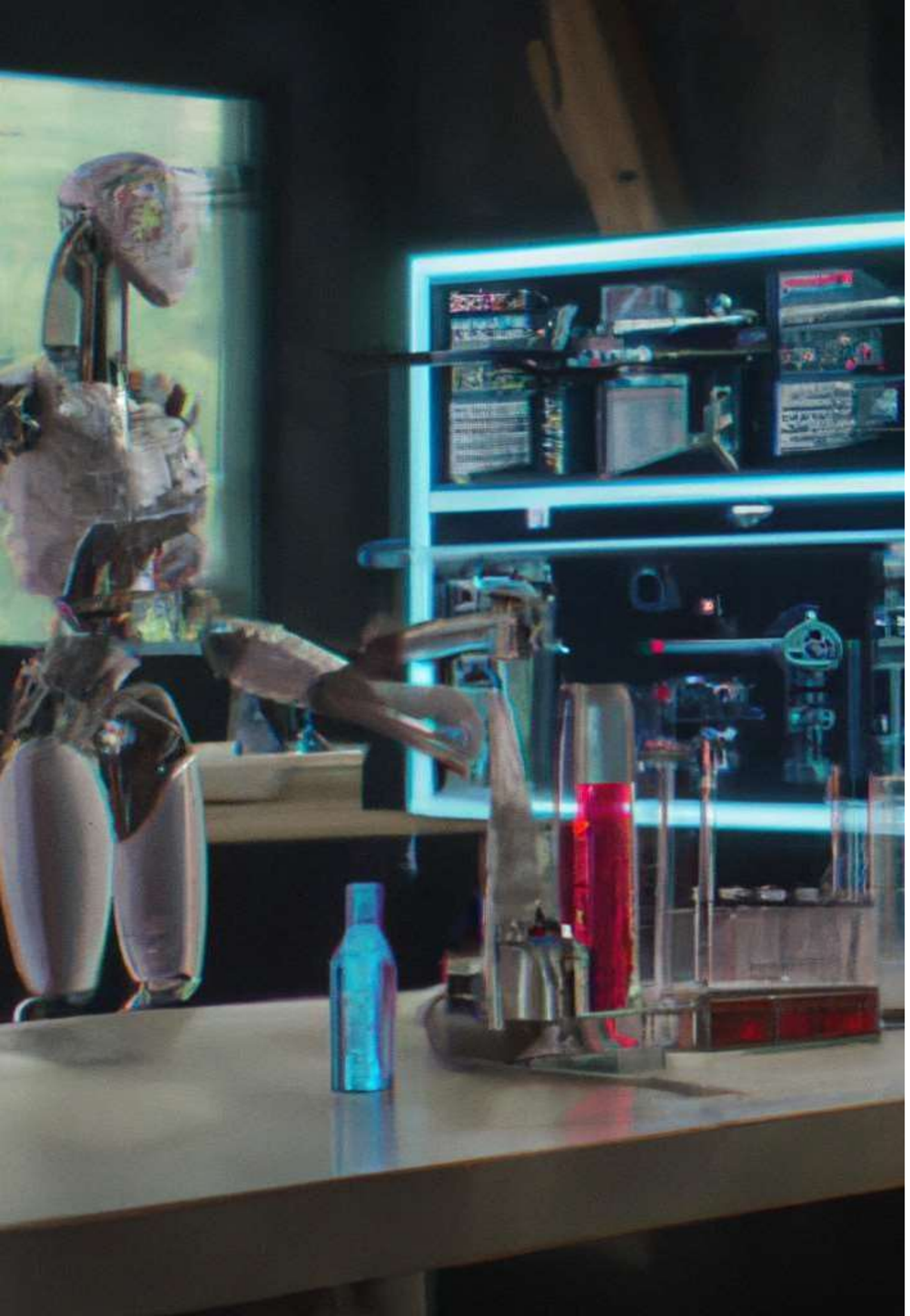
```
data=pd.concat(frames)
```

```
print(data)
```

```
                                        job_title  \
0                                        DevSecOps
0                                Technology Talent
0                                DevSecOps Engineer
0                        New technologies specialist
0                    Data Management Solution Delivery
..                                             ...
0              Développement outil monitoring risques
1   utilisant différentes données textuelles colle...
2   Réflexion stratégique cadre conception platefo...
3           Evaluation bilan carbone d'une entreprise
4   Conception pilote et tokenisation crédit carbo...

                                       skill_required  \
0   software engineering, data development solutio...
0   Broad spectrum technologies (including Blockch...
0   Collaboration, architecture, design, deploymen...
0   Kafka, Java8 (JDK11/17), Spring Cloud Stream, ...
0   Cloudera/Spark/SnowFlake/Databricks/Stratio/Hi...
..                                                ...
0   implémentation chatbot intelligent, outil préd...
1   Les techniques de webscrapping ou/et parsing d...
2
3        méthodologie, questionnaire (questions clés)
4

                                           location  \
0                             SANTANDERTECHHUB-PROFILES
0   Poland, Portugal, Spain, UK, Mexico, Brazil an...
0                                   SANTANDERTECHHUB
0                                      Not specified
0                                      SantanderTechHub
```

DATA PREPARATION

04



Data Preparation :

01 Students Dataset

02 Job Offers Dataset

01 Data Preparation for Students Dataset:

	_id	URL	job_title	location	Name	company	skills	experiences
0	642a399ef246b83896bb9f04	https://www.linkedin.com/in/talel-kb/	Web developer & software engineering student	Gouvernorat Tunis, Tunisie	Talel Kbaier	Triweb	Analyse de données, Compétences analytiques, F...	Développeur web:janv. 2023 - aujourd'hui · 2 m...
1	642a399ef246b83896bb9f05	https://www.linkedin.com/in/omar-talbi-sfax/	Freelance Web Developer	Gouvernorat Sfax, Tunisie	Omar Talbi	Freelance	Laravel, React, JavaScript, PHP, Microsoft Bot...	Freelance Web Developer:août 2019 - aujourd'hu...
2	642a399ef246b83896bb9f06	https://www.linkedin.com/in/hassen-knani-21991...	Web developer chez ZETABOX	Gouvernorat Sfax, Tunisie	Hassen Knani	GOMYCODE	Management, Service client, Microsoft Office, ...	React Js Instructor:mars 2022 - aujourd'hui · ...
3	642a399ef246b83896bb9f07	https://www.linkedin.com/in/safwendammak/	Freelance Web Developer	Gouvernorat Sfax, Tunisie	Safwen Dammak	PixiMind	Angular, PHP/Symfony, Développement web, Dével...	Web Developer:oct. 2020 - févr. 2022 · 1 an 5 ...
4	642a399ef246b83896bb9f08	https://www.linkedin.com/in/oumayma-b%C3%A9hi-...	web & mobile developer 💻	Tunis, Gouvernorat Tunis, Tunisie	Oumayma Béhi	SIGA	NaN	Projet de fin d'études:févr. 2022 - juin 2022 ...
...	...	...	...	...	...	...	...	...
459	642aea77f246b83896bba0d9	https://www.linkedin.com/in/khaoula-ben-othman...	Data Science Engineer	Tunisia	khaoula Ben othman	NaN	MongoDB, RStudio, Business Intelligence (BI),	nan:nan,nan:nan

Entrée [454]:
```
df
```

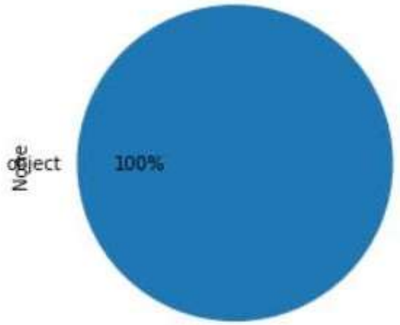
Out[454]:

	URL	job_title	location	Name	company	skills	experiences
0	https://www.linkedin.com/in/talel-kb/	Web developer & software engineering student	Gouvernorat Tunis, Tunisie	Talel Kbaier	Triweb	Analyse de données, Compétences analytiques, F...	Développeur web:janv. 2023 - aujourd'hui · 2 m...
1	https://www.linkedin.com/in/omar-talbi-sfax/	Freelance Web Developer	Gouvernorat Sfax, Tunisie	Omar Talbi	Freelance	Laravel, React, JavaScript, PHP, Microsoft Bot...	Freelance Web Developer:août 2019 - aujourd'hu...
2	https://www.linkedin.com/in/hassen-knani-21991...	Web developer chez ZETABOX	Gouvernorat Sfax, Tunisie	Hassen Knani	GOMYCODE	Management, Service client, Microsoft Office, ...	React Js Instructor:mars 2022 - aujourd'hui · ...
3	https://www.linkedin.com/in/safwendammak/	Freelance Web Developer	Gouvernorat Sfax, Tunisie	Safwen Dammak	PixiMind	Angular, PHP/Symfony, Développement web, Dével...	Web Developer:oct. 2020 - févr. 2022 · 1 an 5 ...
4	https://www.linkedin.com/in/oumayma-b%C3%A9hi-...	web & mobile developer 💻	Tunis, Gouvernorat Tunis, Tunisie	Oumayma Béhi	SIGA	NaN	Projet de fin d'études:févr. 2022 - juin 2022 ...
...	...	...	...	...	...	...	...
459	https://www.linkedin.com/in/khaoula-ben-othman...	Data Science Engineer	Tunisia	khaoula Ben othman	NaN	MongoDB, RStudio, Business Intelligence (BI), ...	nan:nan,nan:nan
460	https://www.linkedin.com/in/farah-abid-989683191/	Data Science Enthusiast	Sfax, Tunisia	Farah Abid	Medicacom	Analyse de données, nlp, Extraction de données...	Data Science Intern:Oct 2022 - Present · 5 mos...
461	https://www.linkedin.com/in/achref-cherif-data...	Data Scientist | Founder & CEO of Data Science...	Tunis, Tunis, Tunisia	Achref Cherif	TEK-UP	Extract, Transform, Load (ETL), Analytical Ski...	Data Scientist:Sep 2020 - Present · 2 yrs 6 mo...

Entrée [395]:
```
df.shape
```

Out[395]: (464, 7)

```
Entrée [397]: df.dtypes.value_counts().plot.pie(autopct='%1.0f%%')
Out[397]: <AxesSubplot:ylabel='None'>
```




```
Entrée [569]: df.describe()
```

Out[569]:

	URL	job_title	location	Name	company	skills	experiences
count	436	435	436	436	380	390	464
unique	435	318	107	435	298	389	378
top	https://www.linkedin.com/in/marwen-hinaoui-479...	Data Science Student	Tunis, Gouvernorat Tunis, Tunisie	Marwen Hinaoui	Freelance	Android, Ionic, React Native, Express.js, Node...	nan:nan,nan:nan
freq	2	24	66	2	16	2	84

```
Entrée [616]: #afficher la liste de pourcentage des valeurs manquantes pour chaque colonne triée par ordre décroissante
              (df.isna().sum()/df.shape[0]).sort_values()
Out[616]: experiences    0.000000
          URL            0.060345
          location       0.060345
          Name           0.060345
          job_title      0.062500
          skills         0.159483
          company        0.181034
          dtype: float64
```

Entrée

Out[

```
Entrée [591]: df= df.drop_duplicates()
```

```
Entrée [592]: df.duplicated().values.any()
Out[592]: False
```

```
Entrée [593]: df.isnull().sum()

Out[593]: URL            1
          job_title      2
          location       1
          Name           1
          company       57
          skills        47
          experiences    0
          dtype: int64
```

```
Entrée [594]: str(df.isna().sum().sum())#total

Out[594]: '109'
```

company

```
Entrée [595]: df[df['company'].isnull()]
```

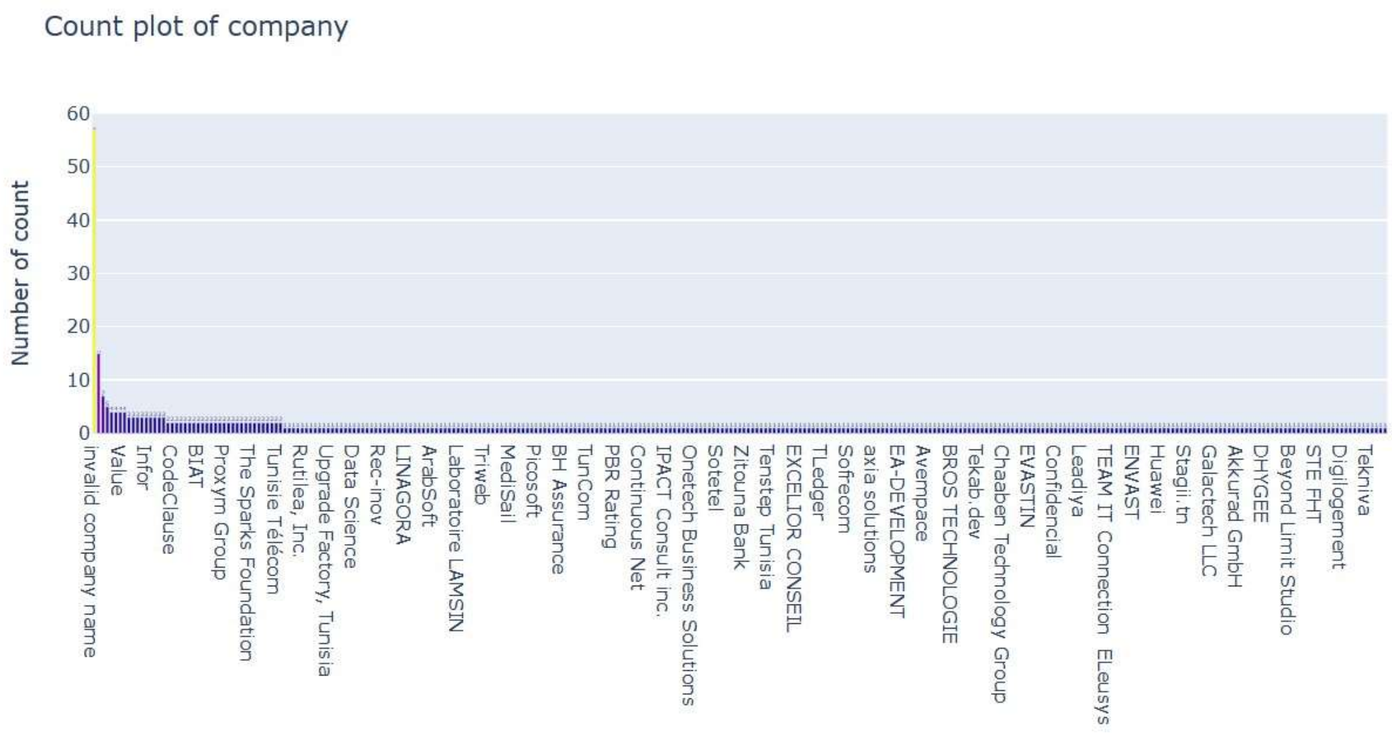
Out[595]:

	URL	job_title	location	Name	company	skills	experiences
5	NaN	NaN	NaN	NaN	NaN	NaN	nan:nan,nan:nan
11	https://www.linkedin.com/in/dhia-boudhraa-243b...	Web developer	Délégation Kairouan Sud, Gouvernorat Kairouan,...	Dhia Boudhraa	NaN	Laravel, Cascading Style Sheets (CSS), HTML5, ...	nan:nan,nan:nan
13	https://www.linkedin.com/in/ramzi-fraj-561814218/	UI/UX Designer Web Developer	Délégation Kélibia, Gouvernorat Nabeul, Tunisie	ramzi fraj	NaN	Anglais, Développement web, Développement d'ap...	nan:nan,nan:nan
23	https://www.linkedin.com/in/imen-ben-yahya-4a7...	Web developer	Gouvernorat Sfax, Tunisie	Imen Ben Yahya	NaN	NaN	nan:nan,nan:nan
28	https://www.linkedin.com/in/marwen-ayedi-1a951...	Front-end web developer:angular||react.	Délégation Sakiet Eddaier, Gouvernorat Sfax, T...	Marwen Ayedi	NaN	Problem Solving, Web Development, Back-End Web...	nan:nan,nan:nan
38	https://www.linkedin.com/in/maryem-waddeni-147...	Full Stack Web Developer	Gouvernorat Nabeul, Tunisie	maryem waddeni	NaN	MySQL, Java, Oracle SQL Developer, Web Design,...	nan:nan,nan:nan

```
Entrée [596]: df['company'].fillna("invalid company name", inplace=True)
```

```
C:\Users\guira\AppData\Local\Temp\ipykernel_16364\3746081923.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
Entrée [597]: df.isnull().sum()

Out[597]: URL            1
          job_title      2
          location       1
          Name           1
          company        0
          skills        47
          experiences    0
          dtype: int64
```



[visualization]

Feature: URL

```
Entrée [626]: df[df['URL'].isnull()]
```

Out[626]:

	URL	job_title	location	Name	company	skills	experiences
5	NaN	NaN	NaN	NaN	invalid company name	NaN	nan:nan,nan:nan

```
Entrée [627]: df = df.drop(5)
```

```
Entrée [629]: df.isnull().sum()
```

Out[629]:

```
URL             0
job_title       1
location        0
Name            0
company         0
skills         46
experiences     0
dtype: int64
```

Feature job_title

Entrée [630]:
```
df[df['job_title'].isnull()]
```

Out[630]:

	URL	job_title	location	Name	company	skills	experiences
149	https://www.linkedin.com/in/ayed-m-085aa9139/	NaN	Mégrine, Gouvernorat Ben Arous, Tunisie	Ayed M.	Octoreality	React Native, Framework Symfony, Laravel, Code...	Mobile Developer:nov. 2022 - aujourd'hui · 4 m...

Entrée [631]:
```
df = df.drop(149)
```

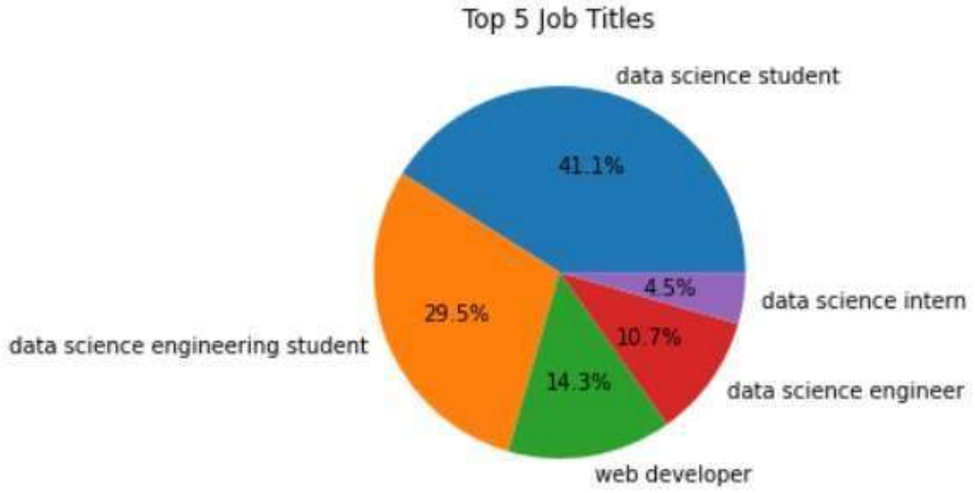
Entrée [632]:
```
df.isnull().sum()
```

Out[632]:
```
URL             0
job_title       0
location        0
Name            0
company         0
skills         46
experiences     0
dtype: int64
```

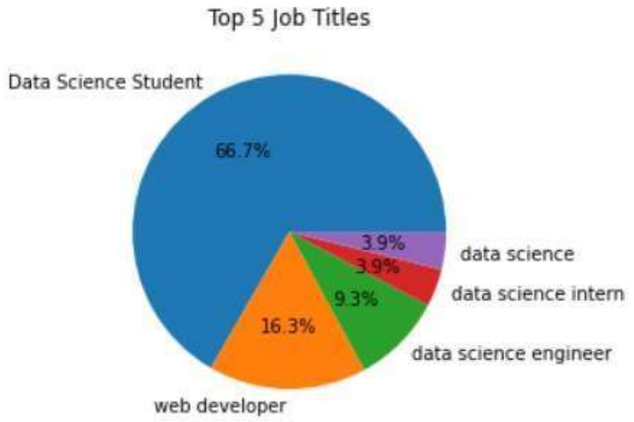
Entrée [1074]:
```
# Convertir toutes les lettres de la colonne "job_title" en minuscules
df['job_title'] = df['job_title'].str.lower()

# Supprimer les caractères de ponctuation dans la colonne job_title
import string
df['job_title'] = df['job_title'].str.translate(str.maketrans('', '', string.punctuation))
```

Entrée [1075]:
```
desc_count1 = df['job_title'].value_counts().reset_index().rename(columns={'index':'index','job_title':'count'})
fig = go.Figure(go.Bar(
    x = desc_count1['index'],y = desc_count1['count'],text=desc_count1['count'],marker={'color': desc_count1['count']}
    ,textposition = "outside"))
fig.update_layout(title_text='Count plot of job_title',xaxis_title="job_title status",yaxis_title="Number of count")
fig.show()
```

```
Entrée [1078]: df['job_title'] = df['job_title'].str.replace('data science student','Data Science Student', case=False)

Entrée [1079]: df['job_title'] = df['job_title'].str.replace('data science engineering student','Data Science Student', case=False)

Entrée [1080]: df['job_title'] = df['job_title'].str.replace('data science student ','Data Science Student', case=False)
```




```
Entrée [1082]: import matplotlib.pyplot as plt

# Get the top 5 job titles and their frequency counts
top_5_job_titles = df['job_title'].value_counts().nlargest(5)

# Create a pie chart of the top 5 job titles
plt.pie(top_5_job_titles.values, labels=top_5_job_titles.index, autopct='%1.1f%%')

# Set the chart title
plt.title('Top 5 Job Titles')

# Display the chart
plt.show()
```



Feature:Skill

```
Entrée [733]: df['skills'].fillna("not mentioned", inplace=True)
```

```
Entrée [291]: new_stopwordsAN = stop_wordsAN
new_stopwordsFR = ['cest','est','dun','en','et','du','des','de']
stopwords_list = stop_wordsFR.union(new_stopwordsAN)
stopwords_list = stopwords_list.union(new_stopwordsFR)
```

```
Entrée [292]: def clean_text(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = re.sub('\n', '', text)
    text = text.replace('-', '')
    text= unidecode.unidecode(text)
    text = ' '.join(word for word in text.split() if word not in stopwords_list)
    return text
```

```
Entrée [293]: df['skills'] = df['skills'].apply(clean_text)
```

```
Entrée [294]: text = " ".join(complaint for complaint in df["skills"])

wordcloud = WordCloud(width = 1200, height = 500,
                background_color ='white',
                 stopwords = stopwords_list,
                min_font_size = 10).generate(text)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

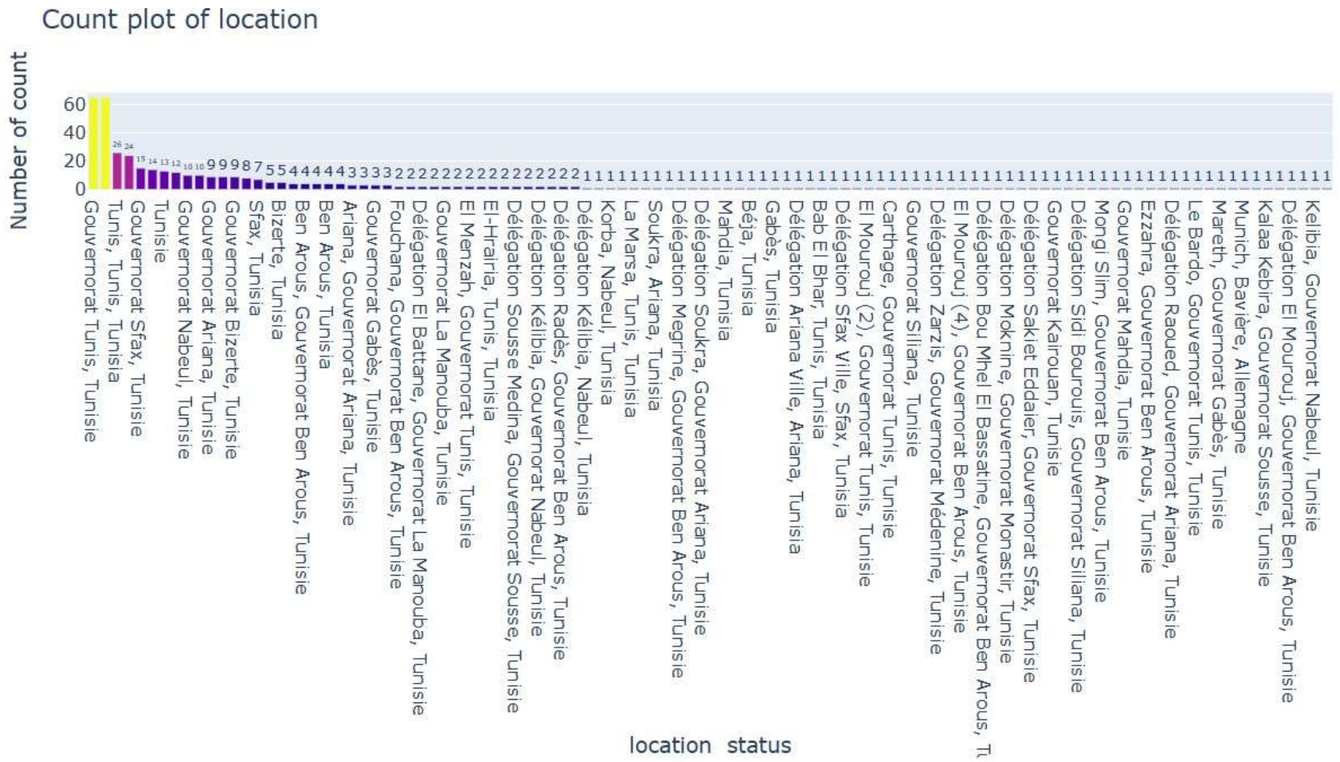
Feature:Location

```
Entrée [286]: df.isnull().sum()
```

```
Out[286]: URL            0
          job_title      0
          location       0
          Name           0
          company        0
          skills         0
          experiences    0
          dtype: int64
```

```
Entrée [288]: str(df.isna().sum().sum())#total
```

```
Out[288]: '0'
```



```
Entrée [1095]: df = df.drop('company', axis=1)
```

```
Entrée [1098]: df
```

Out[1098]:

	URL	job_title	Name	skills	experiences
0	https://www.linkedin.com/in/talel-kb/	web developer software engineering student	Talel Kbaier	analyse donnees competences analytiques framew...	Développeur web:janv. 2023 - aujourd'hui · 2 m...
1	https://www.linkedin.com/in/omar-talbi-sfax/	freelance web developer	Omar Talbi	laravel react javascript php microsoft bot fra...	Freelance Web Developer:août 2019 - aujourd'hu...
2	https://www.linkedin.com/in/hassen-knani-21991...	web developer chez zetabox	Hassen Knani	management service client microsoft office ven...	React Js Instructor:mars 2022 - aujourd'hui · ...
3	https://www.linkedin.com/in/safwendammak/	freelance web developer	Safwen Dammak	angular phpsymfony developpement web developpe...	Web Developer:oct. 2020 - févr. 2022 · 1 an 5 ...
4	https://www.linkedin.com/in/oumayma-b%C3%A9hi-...	web mobile developer 💻	Oumayma Béhi	mentioned	Projet de fin d'études:févr. 2022 - juin 2022 ...
...	...	...	...	...	...
459	https://www.linkedin.com/in/khaoula-ben-othman...	data science engineer	khaoula Ben othman	mongodb rstudio business intelligence bi pyspa...	nan:nan,nan:nan
460	https://www.linkedin.com/in/farah-abid-989683191/	data science enthusiast	Farah Abid	analyse donnees nlp extraction donnees python ...	Data Science Intern:Oct 2022 - Present · 5 mos...
461	https://www.linkedin.com/in/achref-cherif-data...	data scientist founder ceo of data science t...	Achref Cherif	extract transform load etl analytical skills d...	Data Scientist:Sep 2020 - Present · 2 yrs 6 mo...
462	https://www.linkedin.com/in/hamza-samaiy/	data science master student works at vermeg	Hamza Samaiy	developpement web javascript php laravel feuil...	Software Programmer:Jan 2023 - Present · 2 mos...
463	https://www.linkedin.com/in/zitouni-amal-286b9...	ingénieur en informatique spécialité data sci...	Zitouni Amal	microsoft azure machine learning apprentissage...	Ingénieur BI:Oct 2022 - Present · 5 mos,Member...

434 rows × 5 columns

Entrée [568]:
```python
import pandas as pd

# Calculer la précision des valeurs manquantes
missing_values_count = df.isnull().sum()
total_cells = np.product(df.shape)
total_missing = missing_values_count.sum()
accuracy = 1 - (total_missing / total_cells)

# Afficher le résultat
print("L'accuracy de la qualité des données est : ", accuracy)
```

```
L'accuracy de la qualité des données est :  0.916564039408867
```

Entrée [1099]:
```python
import pandas as pd

# Calculer la précision des valeurs manquantes
missing_values_count = df.isnull().sum()
total_cells = np.product(df.shape)
total_missing = missing_values_count.sum()
accuracy = 1 - (total_missing / total_cells)

# Afficher le résultat
print("L'accuracy de la qualité des données est : ", accuracy)
```

```
L'accuracy de la qualité des données est :  1.0
```

02 Data Preparation for Job Offers Dataset:

```
data.shape
```

```
(1593, 5)
```

```
data.columns
```

```
Index(['Unnamed: 0', 'job_title', 'skill_required', 'location', 'link'], dtype='object')
```

removing unnamed column

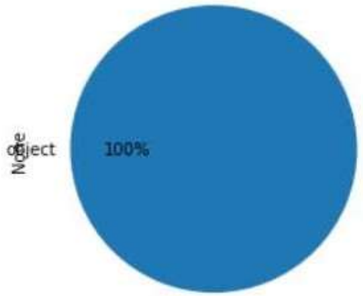
```
df.head()
```

	job_title	skill_required	location	link
0	DevSecOps	software engineering, data development solutio...	SANTANDERTECHHUB-PROFILES	2022-11-11 Job description junior positions-pa...
1	Technology Talent	Broad spectrum technologies (including Blockch...	Poland, Portugal, Spain, UK, Mexico, Brazil an...	2022-11-11 Job description junior positions-pa...
2	DevSecOps Engineer	Collaboration, architecture, design, deploymen...	SANTANDERTECHHUB	2022-11-11 Job description junior positions-pa...
3	New technologies specialist	Kafka, Java8 (JDK11/17), Spring Cloud Stream, ...	Not specified	2022-11-11 Job description junior positions-pa...
4	Data Management Solution Delivery	Cloudera/Spark/SnowFlake/Databricks/Stratio/Hi...	SantanderTechHub	2022-11-11 Job description junior positions-pa...

```
df.duplicated().values.any()
```

True

```
Entrée [397]: df.dtypes.value_counts().plot.pie(autopct='%1.0f%%')
Out[397]: <AxesSubplot:ylabel='None'>
```




```
df.drop_duplicates(inplace=True)
```

```
df.duplicated().values.any()
```

False

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1593 entries, 0 to 1592
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   job_title       1572 non-null   object
 1   skill_required  1398 non-null   object
 2   location        1179 non-null   object
 3   link            1593 non-null   object
dtypes: object(4)
memory usage: 49.9+ KB
```

```
# Vérifier les valeurs manquantes
df.isnull().sum()
```

```
job_title          21
skill_required    195
location          414
link                0
dtype: int64
```

```
#nb de valeurs manquantes
str(df.isna().sum().sum())#total
```

```
'630'
```

```
#afficher la liste de pourcentage des valeurs manquantes
(df.isna().sum()/df.shape[0]).sort_values()
```

```
link              0.000000
job_title         0.013300
skill_required    0.123496
location          0.262191
dtype: float64
```

- **Translation**

```
]: from googletrans import Translator

translator = Translator()



# create a function to translate a single string value
def translate_text(text):
    translation = translator.translate(text)
    return translation.text

# apply the translation function to all cells of the dataframe
df.iloc[:, :-1] = df.iloc[:, :-1].applymap(lambda x: translate_text(x))

# display the translated dataframe
print(df)
```

Développement outil monitoring risques	implémentation chatbot intelligent, outil préd...
utilisant différentes données textuelles colle...	Les techniques de webscrapping ou/et parsing d...
Réflexion stratégique cadre conception platefo...	NaN
Evaluation bilan carbone d'une entreprise	méthodologie, questionnaire (questions clés)

```
                                                job_title  \
                                                DevSecOps
                                          Technology Talent
                                         DevSecOps Engineer
                                  New technologies specialist
                               Data Management Solution Delivery
                                                      ...
                                    Analysis of application logs
                                      Build a reference database
                                  Development of risk monitoring tool
                                  using different collected textual data
                                          Carbon assessment of a company

                                                                skill_required
                                  software engineering, data development solutio...
                                  Broad spectrum technologies (including Blockch...
                                  Collaboration, architecture, design, deploymen...
                                  Kafka, Java8 (JDK11/17), Spring Cloud Stream, ...
                                  Cloudera/Spark/SnowFlake/Databricks/Stratio/Hi...
                                                                                ...
                                  to identify operational anomalies, performance...
                                  on the IS, document and produce the common dat...
                                  intelligent chatbot implementation, predictive...
                                  Techniques for webscrapping and/or parsing pdf...
                                          methodology, questionnaire (key questions)
```

- **replacing missing values with 'NaN'**

```
df.replace(to_replace = {'Unknown':np.nan,'not specified':np.nan,'Not mentioned':np.nan,'N/A':np.nan,'Non spécifié':np.nan,'Not specified':np.nan,'None':np.nan,'Undefined':np.
```

```
df['job_title'].fillna(np.nan, inplace=True)
df['skill_required'].fillna(np.nan, inplace=True)
df['location'].fillna('not specified', inplace=True)
```

- **dropping rows with '-' and '1' values**

```
df=df.drop(df[df['skill_required']=='-'].index)
df=df.drop(df[df['job_title']=='1'].index)
df=df.drop(df[df['skill_required']=='1'].index)
df=df.drop(df[df['job_title']=='-'].index)
```

```
df.isnull().sum()
```

```
job_title          29
skill_required    277
location            0
link                0
```

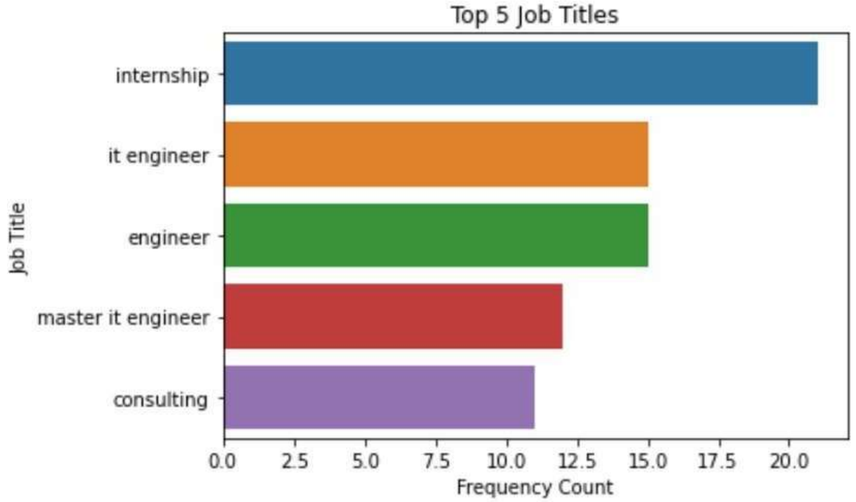
job_title:


```
df=df.dropna(subset=['job_title'])
```

```
df[df['job_title'].isnull()]
```

job_title	skill_required	location	link

```
top_5_job_titles = df['job_title'].value_counts().nlargest(6)
top_5_job_titles
```



skill_required:

```
df=df.dropna(subset=['skill_required'])
```

```
str(df['skill_required'].isna().sum())
```

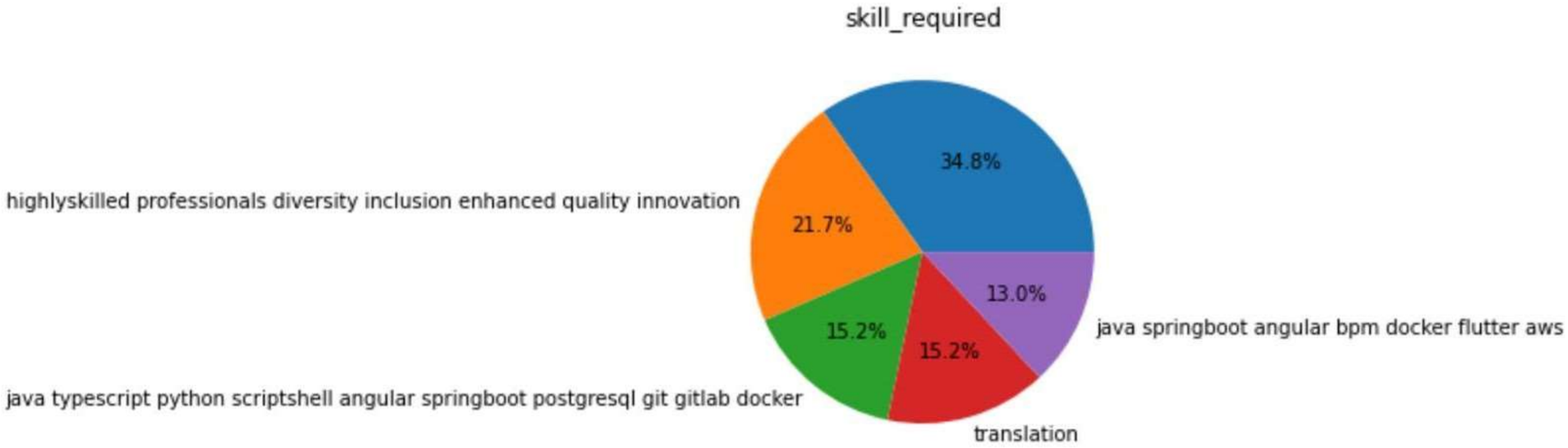
```
'0'
```


```
text = " ".join(complaint for complaint in df["skill_required"])

wordcloud = WordCloud(width = 1200, height = 500,
                background_color ='white',
                  stopwords = stopwords_list,
                min_font_size = 10).generate(text)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

skill_required:

```
top_5_skill_required = df['skill_required'].value_counts().nlargest(6)
top_5_skill_required
```

```
                                                                                16
highlyskilled professionals diversity inclusion enhanced quality innovation     10
java typescript python scriptshell angular springboot postgresql git gitlab docker  7
translation                                                                      7
java springboot angular bpm docker flutter aws                                   6
mechanical engineering aviation industry knowledge                               6
Name: skill_required, dtype: int64
```



```
df.isnull().sum()
```

```
job_title         0
skill_required    0
location          0
link              0
dtype: int64
```

```
str(df.isna().sum().sum())
```

```
'0'
```

```
df
```

	job_title	skill_required	location	link
0	DevSecOps	software engineering, data development solutio...	SANTANDERTECHHUB-PROFILES	2022-11-11 Job description junior positions-pa...
1	Technology Talent	Broad spectrum technologies (including Blockch...	Poland, Portugal, Spain, UK, Mexico, Brazil an...	2022-11-11 Job description junior positions-pa...
2	DevSecOps Engineer	Collaboration, architecture, design, deploymen...	SANTANDERTECHHUB	2022-11-11 Job description junior positions-pa...
3	New technologies specialist	Kafka, Java8 (JDK11/17), Spring Cloud Stream, ...	not specified	2022-11-11 Job description junior positions-pa...
4	Data Management Solution Delivery	Cloudera/Spark/SnowFlake/Databricks/Stratio/Hi...	SantanderTechHub	2022-11-11 Job description junior positions-pa...
...	...	...	...	...
1586	Analyse des logs applicatifs	pour identifier des anomalies de fonctionnemen...	PFE-11	Value-page7.pdf
1587	Construire une base de données de référence	globale sur le SI, documenter et produire le m...	PFE-12	Value-page7.pdf
1588	Développement outil monitoring risques	implémentation chatbot intelligent, outil préd...	not specified	Value-page8.pdf
1589	utilisant différentes données textuelles colle...	Les techniques de webscrapping ou/et parsing d...	not specified	Value-page8.pdf
1591	Evaluation bilan carbone d'une entreprise	méthodologie, questionnaire (questions clés)	not specified	Value-page8.pdf

1274 rows × 4 columns

MODELING

05

Matching and recommandation condidates

```
[33]: import pandas as pd
      import spacy
```

```
[34]: # charger les données
      data1 = pd.read_csv(r"C:\Users\MY NET\Downloads\scraping.csv")
      data2 = pd.read_excel("df_noNA.xlsx")
```

```
[65]: from fuzzywuzzy import fuzz
```

```
[78]: def find_recommendations1(job_title):
          # Sélectionner les compétences requises pour le poste de job_title en data2
          required_skills = data2.loc[data2["job_title"] == job_title]["skill_required"].values[0]

          # Calculer le score de similarité entre les compétences requises et les compétences de chaque job en data1
          scores = data1["skills: "].apply(lambda x: fuzz.token_set_ratio(required_skills, x))
          data1["matching_score"] = scores

          # Trier les jobs en data1 selon le score de similarité décroissant
          sorted_data1 = data1.sort_values(by="matching_score", ascending=False)

          # Afficher les informations requises pour les 5 premiers jobs recommandés
          for index, row in sorted_data1.head(5).iterrows():
              print("Pour le 'Job Title':", job_title)
              print("Mr/Mme:", row["Name"])
              print(" Avec Experiences:", row[" experiences: "])
              print("Son Lien Linkedin est le suivant:", row["URL: "])
              print("Sa Location:", row["location"])
              print("Avec un Matching Score :", row["matching_score"])
              print()
```

```
[80]: find_recommendations1("DevSecOps Engineer")
```

```
Pour le 'Job Title': DevSecOps Engineer
Mr/Mme: Oussama HENI
 Avec Experiences: Full Stack Engineer
Son Lien Linkedin est le suivant: https://tn.linkedin.com/in/oussama-heni
Sa Location: Gouvernorat Ariana, Tunisie
Avec un Matching Score : 41

Pour le 'Job Title': DevSecOps Engineer
Mr/Mme: Hamdi Fhal
 Avec Experiences: nan
Son Lien Linkedin est le suivant: https://tn.linkedin.com/in/hamdi-fhal-a6b5121b2/en?trk=people-guest_people_search-card
Sa Location: Gouvernorat Tunis, Tunisie
Avec un Matching Score : 38

Pour le 'Job Title': DevSecOps Engineer
Mr/Mme: Hamza Arfaoui
 Avec Experiences: nan
Son Lien Linkedin est le suivant: https://tn.linkedin.com/in/hamza-arfaoui-91970b124
Sa Location: Gouvernorat Ariana, Tunisie
Avec un Matching Score : 35

Pour le 'Job Title': DevSecOps Engineer
Mr/Mme: Taha Touzri
 Avec Experiences: Python Developer, Python JavaScript developer, Test Automation Engineer, Python Developer, Research Assistan
t, Software Test Engineer
Son Lien Linkedin est le suivant: https://tn.linkedin.com/in/taha-touzri-274793230
```


- In this code, we will use the fuzz.token_set_ratio() function of the FuzzyWuzzy module of Python
- to calculate the similarity score. This function uses the similarity algorithm Levenshtein character sequence
- to measure the similarity between two character strings.
- The similarity score is a measure of the similarity between the two strings, ranging from 0 (no similarity) to 100 (perfect match).

Matching and recommanding job offer

```
[35]: import pandas as pd
      from fuzzywuzzy import fuzz
```

```
[36]: # charger les données
      data1 = pd.read_csv(r"C:\Users\MY NET\Downloads\scraping.csv")
      data2 = pd.read_excel("df_noNA.xlsx")
```

```
                       _id                Name             link  \
642b57b2338e229c8c9bb4d8     Mohamed BOURAOUI  Value-page8.pdf
642b57b2338e229c8c9bb4d9   Abdelkader Dhouibi  Value-page8.pdf
642b57b2338e229c8c9bb4da        Mahmoud Segni  Value-page8.pdf
642b57b2338e229c8c9bb4db          Taha Touzri  Value-page8.pdf
642b57b2338e229c8c9bb4dc      Mahmoud Aloulou  Value-page8.pdf
                      ...                 ...              ...
642b722e7624402b2227812e        Bassem Gharbi  Value-page8.pdf
642b722e7624402b2227812f         Oussama HENI  Value-page8.pdf
642b722e7624402b22278130        Hamza Arfaoui  Value-page8.pdf
642b722e7624402b22278131          Bacem Smiri  Value-page8.pdf
642b722e7624402b22278132  Mohamed Amine SALAH  Value-page8.pdf

                                    job_title matching_score
[Implementation process management APP Interns...            100
[Integration Internship, Azure Devops Internsh...            100
[Mobile application designer and developer, IA...            100
                  [IA, DataScience, Kotlin, Java]            100
[Buisness Analyst, QA, Data Scientist, IT Busi...            100
                                        ...                  ...
[Implementation process management APP Interns...            100
[Flutter Mobile Developers, Développeur Front-...            100
[Implementation process management APP Interns...            100
[Consultant Engineer, Business analyst, C# Pro...             82
[Mobile application designer and developer, IA...            100
```

```
[40]: data2.columns
```

```
[40]: Index(['Unnamed: 0', 'job_title', 'skill_required', 'location', 'link'], dtype='object')
```

```
[42]: def find_recommendations():
          recommendations = pd.DataFrame(columns=['_id', 'Name', 'link', 'job_title', 'matching_score'])
          for index, row in data1.iterrows():
              matching_scores = []
              for idx, job in data2.iterrows():
                  matching_score = fuzz.token_set_ratio(row['skills: '], job['skill_required'])
                  if matching_score > 80:
                      matching_scores.append((job['job_title'], matching_score))
              matching_scores.sort(key=lambda x: x[1], reverse=True)
              recommended_jobs = [job[0] for job in matching_scores]
              matching_score = matching_scores[0][1] if matching_scores else 0
              recommendations = recommendations.append({'_id': row['_id'], 'Name': row['Name'], 'link': job['link'], 'job_title': rec
          return recommendations
```

```
[43]: # Affichage des recommandations
      recommendations = find_recommendations()
      top_recommendations = recommendations.groupby('_id').apply(lambda x: x.sort_values(by="matching_score", ascending=False).head(1)
      top_recommendations = top_recommendations.reset_index(drop=True)[['_id', 'Name', 'link', 'job_title', 'matching_score']]
      print(top_recommendations)
```



```
[47]: def get_recommended_job_titles1(_id):
          row = data1.loc[data1['_id'] == _id].iloc[0]
          matching_scores = []
          for idx, job in data2.iterrows():
              matching_score = fuzz.token_set_ratio(row['skills: '], job['skill_required'])
              if matching_score > 80:
                  matching_scores.append((job['job_title'], job['link'], matching_score))
          matching_scores.sort(key=lambda x: x[2], reverse=True)
          recommended_jobs = [{'job_title': job[0], 'link': job[1]} for job in matching_scores]
          return recommended_jobs
```

```
[59]: recommended_jobs = get_recommended_job_titles1('642b57b2338e229c8c9bb4d8')
      for job in recommended_jobs:
        print('Le job title recommendé:', job['job_title'], ' Son LIEN:', job['link'])
```

```
Le job title recommendé: Implementation process management APP Internship  Son LIEN: DATAHORIZON-page17.pdf
Le job title recommendé: Ingénieur Data Scientist  Son LIEN: EY-page86.pdf
Le job title recommendé: Développeur de Reconnaissance de recherche  Son LIEN: GENITECH-page7.pdf
Le job title recommendé: Notification System Administrator  Son LIEN: Keyrus-page12.pdf
Le job title recommendé: équipe  Son LIEN: pfe-book-Aymax (1)-page17.pdf
Le job title recommendé: Data Scientist Engineer  Son LIEN: Value-page20.pdf
Le job title recommendé: Engineer  Son LIEN: Value-page21.pdf
```

Matching and recommandation for condidates and job offer

removing stop words from skill_required column

[9]:
```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

nltk.download('stopwords')
nltk.download('punkt')
```

```
[nltk_data] Downloading package stopwords to C:\Users\MSI
[nltk_data]     GF63\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to C:\Users\MSI
[nltk_data]     GF63\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

[9]:
```
True
```

[10]:
```
stop_words = set(stopwords.words('english'))

def clean(text):
    tokens = word_tokenize(text)

    filtered_tokens = [word for word in tokens if not word.lower() in stop_words]

    filtered_text = ' '.join(filtered_tokens)
    return filtered_text
```

[11]:
```
df1['skill_required'] = df1['skill_required'].apply(clean)
df2['skills'] = df2['skills'].apply(clean)
```

7]:
```
print(df1['skill_required'][4])
```

```
Cloudera/Spark/SnowFlake/Databricks/Stratio/Hive/Impala/HBase , Spark Streaming/Flink/Storm , API Ecosystems , HDFS , S3 , Kafka , Scala/Python/J
CI/CD , Google/AWS/Azure
```

8]:
```
print(df2['skills'][3])
```

```
angular phpsymfony developpement web developpement logiciels applications web conception technique methodes agiles gestion projet developpement f
d javascript jquery php wordpress xml json plsql uml html symfony cakephp typescript developpement web backend framework symfony conception front
t mysql sql twitter bootstrap ajax feuilles style cascade css linux api rest
```

2]:
```
import ast
#list_from_string = ast.literal_eval("['Python', 'AI', 'Digitallending']")

#print(list_from_string)
def tolist(text):
    try:
        split_skills= ast.literal_eval(text)
        split_skills = [item.split('/') for item in split_skills]
        flat_list = [item.strip() for sublist in split_skills for item in sublist]
        return flat_list
    except:
        split_skills=text.split(',')
        split_skills = [item.split('/') for item in split_skills]
        flat_list = [item.strip() for sublist in split_skills for item in sublist]
        return flat_list


df1['skill_required'] = df1['skill_required'].apply(tolist)
df2['skills'] = df2['skills'].apply(tolist)
```

[19]:
```
def lemmatize_stemming(text):
    return stemmer.stem(WordNetLemmatizer().lemmatize(text, pos='v'))
def preprocess(text):
    result = []
    text=str(text)
    # Remove stopwords from the text
    text = remove_stopwords(text)
    for token in gensim.utils.simple_preprocess(text):
        if token not in gensim.parsing.preprocessing.STOPWORDS and len(token) > 3:
            if token == 'xxxx':
                continue
            result.append(lemmatize_stemming(token))

    return result
```

[20]:
```
processed_docs=df1['skill_required'].map(preprocess)
```

[21]:
```
p=df2['skills'].map(preprocess)
```

[22]:
```
print(p[5])
```

```
['program', 'languag', 'microsoft', 'excel', 'javascript', 'program', 'program', 'concept
ri', 'cascad', 'style', 'sheet', 'python', 'program', 'languag', 'adob', 'photoshop', 'ex
'illustr', 'graphism', 'standard', 'bootstrap', 'developp', 'developp', 'frontend', 'fron
cess', 'access', 'typographi', 'typographi', 'site', 'adaptatif', 'sass', 'stylesheet',
```

[23]:
```
print(processed_docs[13])
```

```
['valorará', 'acreditar', 'experiencia', 'profesion', 'buena', 'soft', 'skill', 'buena',
'miedo', 'cambio', 'orientado', 'objetivo', 'buena', 'habilidad', 'ofimática', 'excel',
```

[24]:
```
print(len(processed_docs))
```

```
1298
```

[24]:
```
print(len(processed_docs))
```

```
1298
```

[25]:
```
print(len(p))
```

```
434
```

[26]:
```
processed_data=pd.concat([processed_docs,p])
```

[27]:
```
processed_data2=processed_data.map(preprocess)
```

[28]:
```
processed_data=pd.concat([processed_data,processed_data2])
```

[29]:
```
print(len(processed_data))
```

```
3464
```

[30]:
```
def word2vec_model():
    w2v_model = Word2Vec(min_count=1,
                         window=3,
                         vector_size=50,
                         sample=6e-5,
                         alpha=0.03,
                         min_alpha=0.0007,
                         negative=20)

    w2v_model.build_vocab(processed_data)
    w2v_model.train(processed_data, total_examples=w2v_model.corpus_count, epochs=300, report_delay=1)

    return w2v_model
```

[31]:
```
w2v_model = word2vec_model()
w2v_model.save('word2vec_model_noSW')
```

[32]:
```
emb_vec = w2v_model.wv
```

[33]:
```
emb_vec['python'] # It will return vector representation of the word python
```

[33]:
```
array([-0.19220068, -1.555042  ,  0.12525329, -0.27510718,  0.28673938,
       -1.2068081 ,  0.15263467, -0.08739781,  0.7810219 ,  0.5298093 ,
        0.40692335,  0.23484743,  0.12515357,  0.32432836, -0.51571864,
        0.30603442,  0.69985527,  0.9677374 , -0.6049835 ,  0.6400751 ,
        0.3115423 , -0.0818011 ,  0.3242386 ,  1.2411157 ,  0.3901614 ,
        1.0849819 ,  0.2465001 ,  0.7385269 ,  0.15495509, -0.06782021,
       -0.22350268, -0.94045323, -0.56122214, -0.5374459 , -1.2871603 ,
        0.5722579 ,  1.622912  , -0.23215348, -0.82555664, -0.17313346,
       -0.3513574 , -0.40873203, -0.53295267,  0.716141  ,  0.09828158,
       -0.5442361 , -0.21533252, -1.1094925 ,  0.05039541, -0.26867712],
      dtype=float32)
```

Cosine Similarity

[34]:
```
def find_similarity(sen1, sen2, model):
    p_sen1 = preprocess(sen1)
    p_sen2 = preprocess(sen2)

    sen_vec1 = np.zeros(50)
    sen_vec2 = np.zeros(50)
    for val in p_sen1:
        sen_vec1 = np.add(sen_vec1, model[val])

    for val in p_sen2:
        sen_vec2 = np.add(sen_vec2, model[val])
    return dot(sen_vec1,sen_vec2)/(norm(sen_vec1)*norm(sen_vec2))
```

[84]:
```
find_similarity('Android, Java', 'Android, Java, Python, Django, Xamarin, Cloud',emb_vec )
```

[84]:
```
0.8696490222185447
```

[36]:
```
find_similarity('Java', 'mobil',emb_vec )
```

[36]:
```
0.42059180604019664
```

[84]:
```
find_similarity('Android, Java', 'Android, Java, Python, Django, Xamarin, Cloud',emb_vec )
```

[84]:
```
0.8696490222185447
```

[36]:
```
find_similarity('Java', 'mobil',emb_vec )
```

[36]:
```
0.42059180604019664
```

[54]:
```
def similarity(skill,required,model):
    p_skill=preprocess(skill)
    p_required=preprocess(required)
    result=[]
    for i in range(len(p_required)):
        similarities=[]
        for j in range(len(p_skill)):
            similarities.append(find_similarity2(p_skill[j],p_required[i],model))
        result.append(max(similarities))
    result = [0 if pd.isna(x) else x for x in result]
    if len(result) > 0:
        avg_similarity = sum(result) / len(result)
    else:
        avg_similarity = 0
    return avg_similarity,required,result
    #we returned the similarity score , the required skills  ,all similiraty for every word alone
```

]:
```
similarity('Android, Java, Python, Django', 'Android, Java ',emb_vec )
```

]:
```
(1.0, 'Android, Java ', [1.0, 1.0000000000000002])
```

]:
```
similarity('Android, Java', 'Android, Java, Python, Django ',emb_vec )
```

]:
```
(0.864017511456643,
 'Android, Java, Python, Django ',
 [1.0, 1.0000000000000002, 0.848408544878991, 0.6076615009475811])
```

Some adjustement to try

[601]:
```
processed_data.duplicated()
```

[601]:
```
0      False
1      False
2      False
3      False
4      False
       ...
429     True
430    False
431    False
432     True
433    False
Length: 3464, dtype: bool
```

[602]:
```
processed_data2=processed_data.drop_duplicates()
```

[603]:
```
len(processed_data2)
```

[603]:
```
2054
```

[604]:
```
def word2vec_model2():
    w2v_model2 = Word2Vec(min_count=1,
                          window=3,
                          vector_size=50,
                          sample=6e-5,
                          alpha=0.03,
                          min_alpha=0.0007,
                          negative=20)

    w2v_model2.build_vocab(processed_data2)
    w2v_model2.train(processed_data, total_examples=w2v_model.corpus_count, epochs=300, report_delay=1)

    return w2v_model2
```

[605]:
```
w2v_model2 = word2vec_model2()
w2v_model2.save('word2vec_model_V2')
```

[606]:
```
emb_vec2 = w2v_model2.wv
```

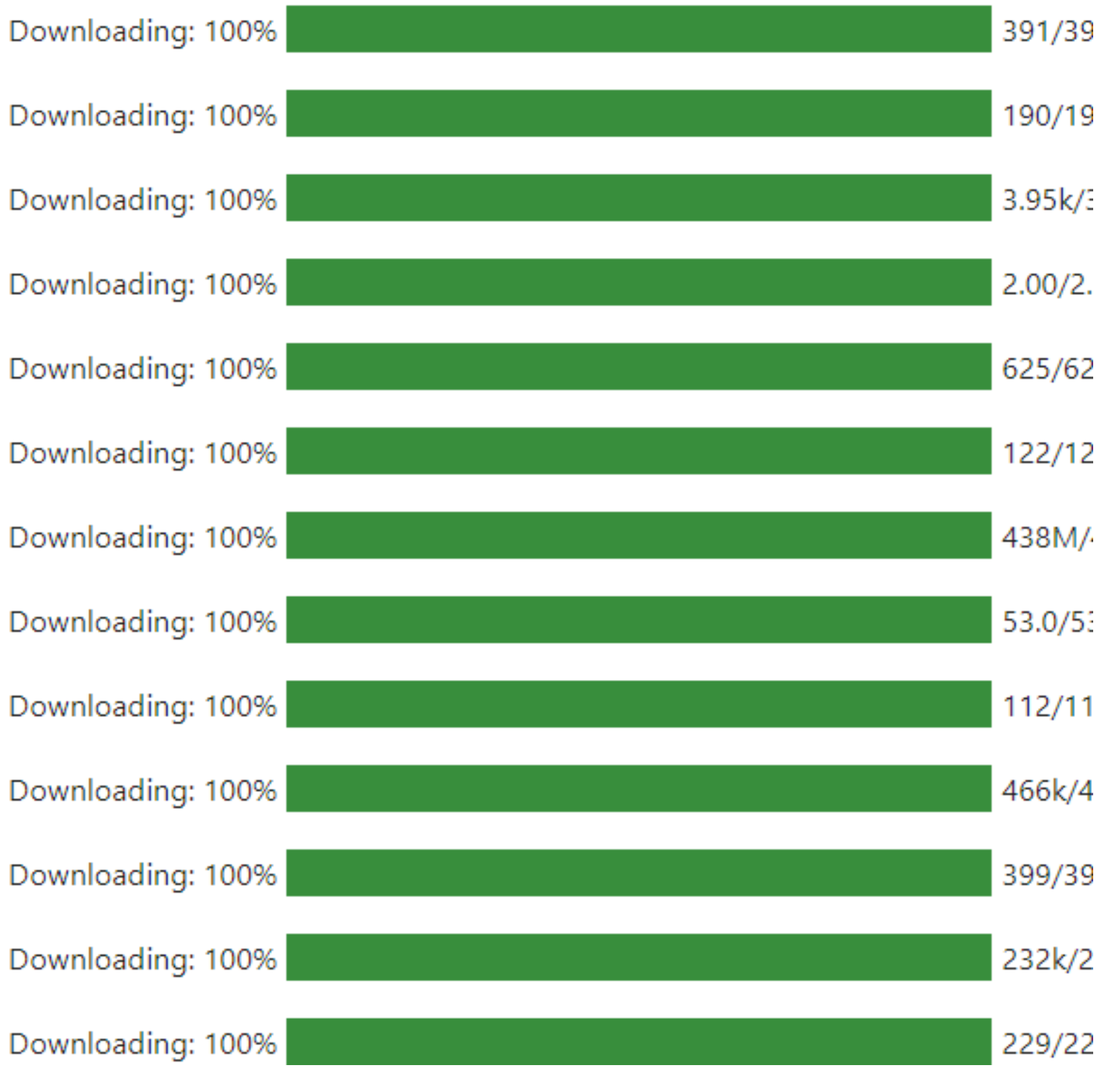
[607]:
```
similarity(df2['skills'][4],df1['skill_required'][2],emb_vec2)#need to train the emb_vec on the skills dataset too
```

[607]:
```
(0.17915858044855756,
 ['Collaboration',
```

```
print(df2['skills'][4])
print(df1['skill_required'][2])
```

```
['mentioned']
['Collaboration', 'architecture', 'design', 'deployment solutions', 'system integration', 'automation', 'execution support', 'pipeline deployments', 'A
LM tools', 'cyber security', 'SW deployments', 'automation functional regression tests', 'continuous integration', 'environments infrastructures', 'con
tinuous monitoring', 'automatic deployment tools ( Terraform', 'Ansible', 'etc . )', 'cloud architectures ( vCloud', 'Azure', 'AWS )', 'monitoring tool
```

```
model = SentenceTransformer('bert-base-nli-mean-tokens')
```




```
BERTsimilarity('Android,Java,Cloud','Java',model)
```

```
(1.0000001192092896, 'Java', [1.0000001])
```

```
BERTsimilarity('Java','Android,Java,Cloud',model)
```

```
(0.7070481777191162, 'Android,Java,Cloud', [0.6796782, 1.0000001, 0.4414662])
```

```
similarity('Java','Android,Java,Cloud',emb_vec)
```

```
(0.6654542758497978,
 'Android,Java,Cloud',
 [0.7962357363258589, 0.9999999999999999, 0.2001270912235344])
```

```
similarity('Java','cloud',emb_vec)
```

```
(0.2001270912235344, 'cloud', [0.2001270912235344])
```

```
BERTsimilarity('Java','cloud',model)
```

```
(0.44146621227264404, 'cloud', [0.4414662])
```

```
BERTsimilarity(df2['skills'][4],df1['skill_required'][2],model)
```

```
(0.7424084604890259,
 ['Collaboration',
  'architecture',
  'design',
  'deployment solutions',
  'system integration',
  'automation',
```

Matching and recommandation for single row

[79]:
```
def match_person(cv_row,jbdf,emb_vec):
    sim_score=0
    for j in range(len(jbdf)):
        s=similarity(cv_row['skills'],jbdf['skill_required'][j] , emb_vec)[0]
        if s>sim_score:
            sim_score, skills_list, score_list = similarity(cv_row['skills'],jbdf['skill_required'][j] , emb_vec)
            row_job=jbdf.iloc[j]
    return row_job,sim_score,skills_list,score_list
```

[87]:
```
def match_job(jb_row,cvdf,emb_vec):
    sim_score=0
    for j in range(len(cvdf)):
        s=similarity(cvdf['skills'][j],jb_row['skill_required'] , emb_vec)[0]
        if s>sim_score:
            sim_score, skills_list, score_list = similarity(cvdf['skills'][j],jb_row['skill_required'] , emb_vec)
            row_job=cvdf.iloc[j]
    return row_job,sim_score,skills_list,score_list
```

[81]:
```
match_person(df2.iloc[1],df1,emb_vec)
```

```
C:\Users\MSI GF63\AppData\Local\Temp\ipykernel_15404\3122937574.py:23: RuntimeWarning: invalid value encountered in double_scalars
  return dot(sen_vec1,sen_vec2)/(norm(sen_vec1)*norm(sen_vec2))
```

[81]:
```
(job_title        Implementation process management APP Internship
 skill_required                                            [Java]
 location                                                     NaN
 link                                        DATAHORIZON-page17.pdf
 Name: 356, dtype: object,
 1.0000000000000002,
 ['Java'],
 [1.0000000000000002])
```

```
match_job(df1.iloc[3],df2,emb_vec)
```

```
C:\Users\MSI GF63\AppData\Local\Temp\ipykernel_15404\3122937574.p
  return dot(sen_vec1,sen_vec2)/(norm(sen_vec1)*norm(sen_vec2))
```

```
(URL                        https://www.linkedin.com/in/nourkobbi/
 job_title                    data science and engineering student
 Name                                                    Nour Kobbi
 skills          [angularjs net framework developpement logicie...
 experiences                                          nan:nan,nan:nan
 Name: 247, dtype: object,
 0.7105570850909376,
 ['Kafka',
  'Java8 ( JDK11',
  '17 )',
  'Spring Cloud Stream',
  'Spring Boot',
  'Spring Cloud ( Web',
  'Data',
  'JPA )',
  'Spring Framework Reactive',
  'WebFlux',
  'Microservices',
  'Event-Driven architecture',
  'MongoDB',
  'Neo4J',
  'SaaS software level',
  'Openshift',
  'K8S',
  'Flink'],
 [1.0,
  1.0000000000000002,
  1.0000000000000002,
  0.36318970446000404,
  0.3722733426152549,
```

```
match_job(df1.iloc[3],df2,emb_vec)
```

```
C:\Users\MSI GF63\AppData\Local\Temp\ipykernel_15404\3122937574.p
  return dot(sen_vec1,sen_vec2)/(norm(sen_vec1)*norm(sen_vec2))
```

```
(URL                        https://www.linkedin.com/in/nourkobbi/
 job_title                    data science and engineering student
 Name                                                    Nour Kobbi
 skills          [angularjs net framework developpement logicie...
 experiences                                          nan:nan,nan:nan
 Name: 247, dtype: object,
 0.7105570850909376,
 ['Kafka',
  'Java8 ( JDK11',
  '17 )',
  'Spring Cloud Stream',
  'Spring Boot',
  'Spring Cloud ( Web',
  'Data',
  'JPA )',
  'Spring Framework Reactive',
  'WebFlux',
  'Microservices',
  'Event-Driven architecture',
  'MongoDB',
  'Neo4J',
  'SaaS software level',
  'Openshift',
  'K8S',
  'Flink'],
 [1.0,
  1.0000000000000002,
  1.0000000000000002,
  0.36318970446000404,
  0.3722733426152549,
```

Matching and recommanding dataset

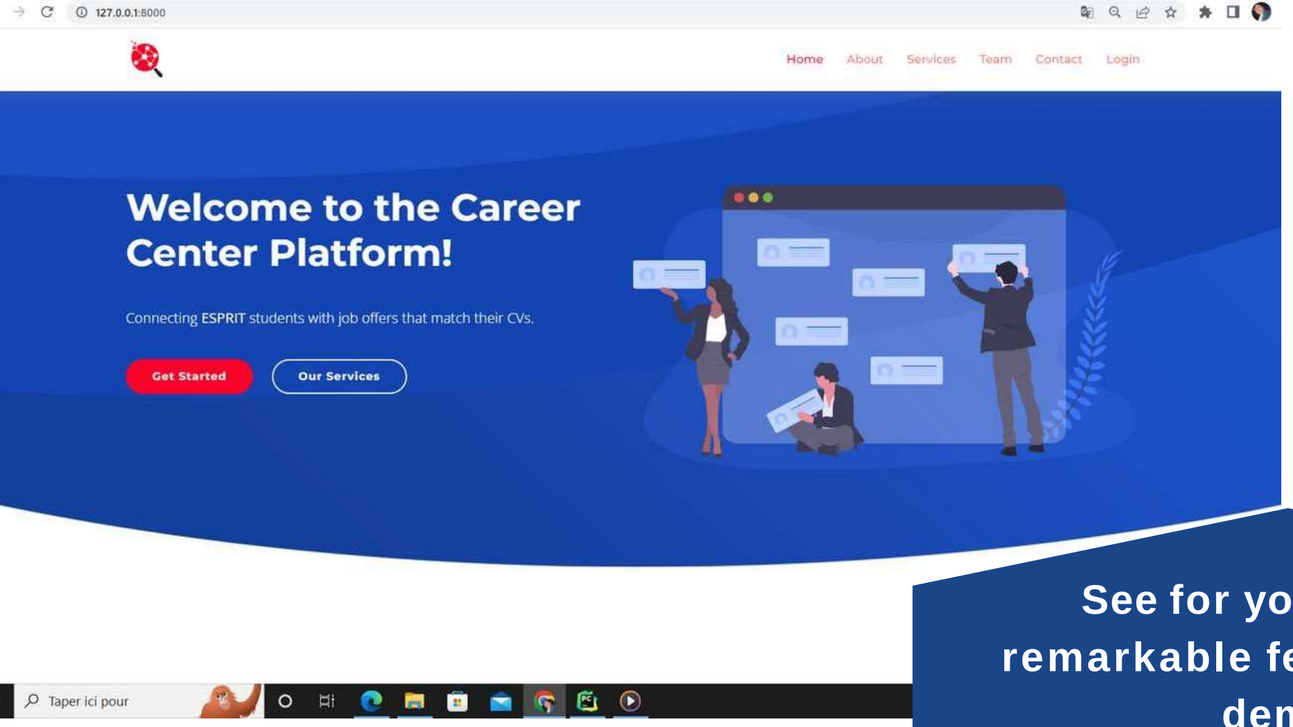
[72]:
```
def Match(cvdf,jbdf,emb_vec):
    matchdf = pd.DataFrame(columns=['URL', 'Name', 'job_matched', 'job skills', 'total_score', 'score_list'])
    for i in range(len(cvdf)):
        print(cvdf.iloc[i])
        sim_score=0
        for j in range(len(jbdf)):
            s=similarity(cvdf['skills'][i],jbdf['skill_required'][j] , emb_vec)[0]
            if s>sim_score:
                sim_score, skills_list, score_list = similarity(cvdf['skills'][i],jbdf['skill_required'][j] , emb_vec)
                row_job=jbdf.iloc[j]
        matchdf = pd.concat([matchdf, pd.DataFrame({
                  'URL': [cvdf['URL'][i]],
                  'Name': [cvdf['Name'][i]],
                  'job_matched': [row_job['job_title']],
                  'job skills': [skills_list],
                  'total_score': [sim_score],
                  'score_list': [score_list]
                  })])

    return matchdf
```

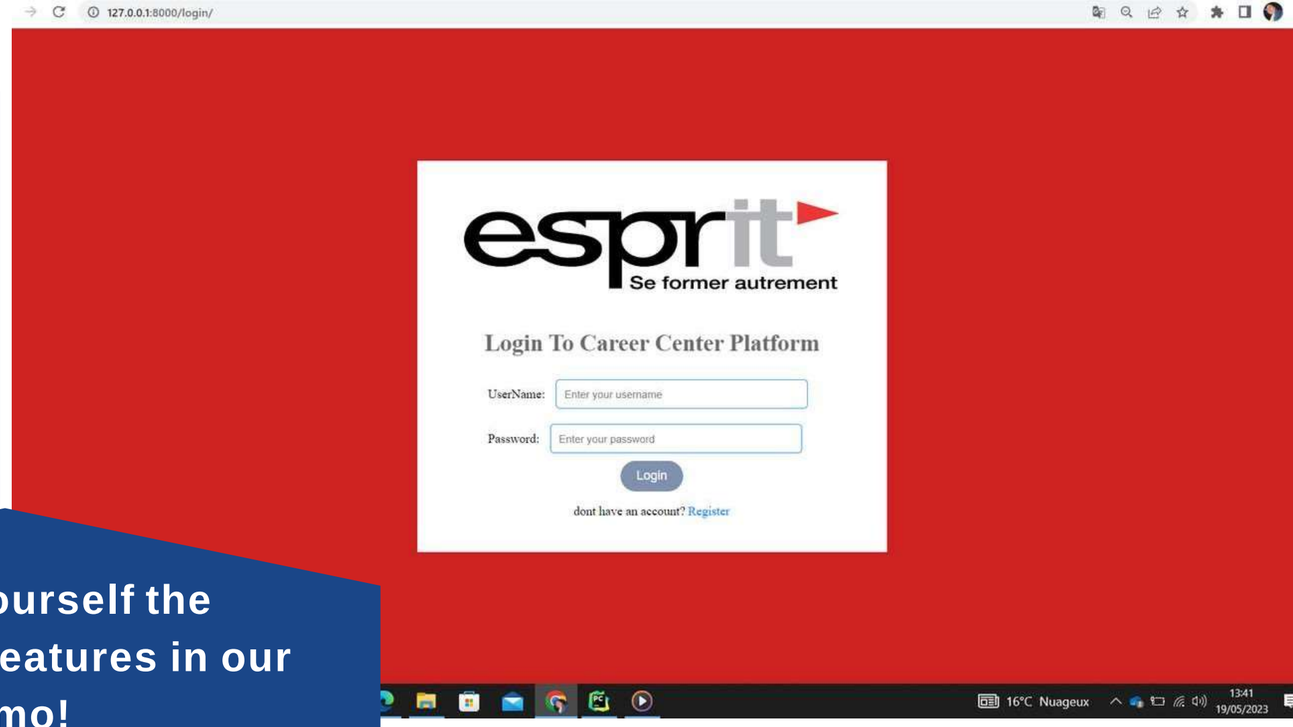
[73]:
```
matchdf=Match(df2,df1,emb_vec)
```

	URL	Name	job_matched	job skills	total_score	score_list
2	https://www.linkedin.com/in/talel-	Talel Kbaier	Implementation process management APP Internsh	['Java']	1	[1.0000000000000002]
3	https://www.linkedin.com/in/oma	Omar Talbi	Implementation process management APP Internsh	['Java']	1	[1.0000000000000002]
4	https://www.linkedin.com/in/hass	Hassen Knani	Master Ingénieur réseau et sécurité	['Projet']	1	[1.0000000000000002]
5	https://www.linkedin.com/in/safw	Safwen Dammak	CONCEPTION ET DÉVELOPPEMENT D'UN SIMULAT	['C++', 'QT', 'LINUX', 'SOME', 'IP', '']	1	[1.0000000000000002]
6	https://www.linkedin.com/in/oum	Oumayma Béhi	innovation specialist	['curiosity', 'creativity', 'adaptability']	0,26507317	[0.2622474085608724, 0.2650820348991975, 0.26789
7	https://www.linkedin.com/in/med-	Med Yassine Ben Romdhane	Trusted Firmware Implementation	['C', 'C++', 'STM32', 'UI Development']	1	[1.0000000000000002]
8	https://www.linkedin.com/in/hous	Houssem Derouich	Développeur Front-End	['Javascript', 'CSS', 'HTML']	0,976186893	[1.0000000000000002, 0.9523737865925803]
9	https://www.linkedin.com/in/abde	Omar Abdelkefi	Implementation process management APP Internsh	['Java']	1	[1.0000000000000002]
10	https://www.linkedin.com/in/med-	Med.Ridha Harhira	candidature	['COMMUNICATION']	1	[1.0000000000000002]
11	https://www.linkedin.com/in/dhia-	Dhia Boudhraa	Trusted Firmware Implementation	['C', 'C++', 'STM32', 'UI Development']	1	[1.0000000000000002]
12	https://www.linkedin.com/in/amin	Amine AZRI	Implementation process management APP Internsh	['Java']	1	[1.0000000000000002]
13	https://www.linkedin.com/in/ramz	ramzi fraj	Mobile and Web Application Developer	['Experience developing mobile web app	0,773672356	[0.3900851649758914, 0.7046042575953507, 0.99999
14	https://www.linkedin.com/in/saou	Saoussen Ben Mohamed	innovation specialist	['curiosity', 'creativity', 'adaptability']	0,26507317	[0.2622474085608724, 0.2650820348991975, 0.26789
15	https://www.linkedin.com/in/marc	Marouen Kouki	Trusted Firmware Implementation	['C', 'C++', 'STM32', 'UI Development']	1	[1.0000000000000002]
16	https://www.linkedin.com/in/ikran	Ikram Nebti	Implementation process management APP Internsh	['Java']	1	[1.0000000000000002]
17	https://www.linkedin.com/in/yessi	Yessin Rebhi	candidature	['COMMUNICATION']	1	[1.0000000000000002]
18	https://www.linkedin.com/in/rouis	Rouissi Adnen	Implementation process management APP Internsh	['Java']	1	[1.0000000000000002]
19	https://www.linkedin.com/in/majd	Majd Ghabri	Master Ingénieur réseau et sécurité	['Projet']	1	[1.0000000000000002]
20	https://www.linkedin.com/in/yassi	Yassin Kaabi	Créer plateforme de gestion et évaluation des taler	['FULLSTACK']	1	[1.0]
21	https://www.linkedin.com/in/oma	Omar Naifar	candidature	['COMMUNICATION']	1	[1.0000000000000002]
22	https://www.linkedin.com/in/aym	Aymen Haji	Développeur Front-End	['Javascript', 'CSS', 'HTML']	1	[1.0000000000000002, 0.9999999999999999]
23	https://www.linkedin.com/in/imen	Imen Ben Yahya	innovation specialist	['curiosity', 'creativity', 'adaptability']	0,26507317	[0.2622474085608724, 0.2650820348991975, 0.26789
24	https://www.linkedin.com/in/moh	Mohamed Bechir Mejri	Trusted Firmware Implementation	['C', 'C++', 'STM32', 'UI Development']	1	[1.0000000000000002]
25	https://www.linkedin.com/in/abidi	Abidi Khaireddine	Implementation process management APP Internsh	['Java']	1	[1.0000000000000002]
26	https://www.linkedin.com/in/saida	Said Atoui	Ingénieur Data Scientist	['Python']	1	[1.0]
27	https://www.linkedin.com/in/marv	Marwen Ayedi	Trusted Firmware Implementation	['C', 'C++', 'STM32', 'UI Development']	1	[1.0000000000000002]
28	https://www.linkedin.com/in/khali	Khalil Bouhdida	Développement outil support SMS	['PHP', 'MySQL']	1	[1.0000000000000002]
29	https://www.linkedin.com/in/louja	Loujaien Limayma	CONCEPTION ET DÉVELOPPEMENT D'UN SIMULAT	['C++', 'QT', 'LINUX', 'SOME', 'IP', '']	1	[1.0000000000000002]
30	https://www.linkedin.com/in/nissri	Nissrine Hnainia	Mobile Application Developer	['React Native', 'NodeJs', 'JavaScript']	0,976936668	[1.0, 0.9077466718105218, 1.0, 1.0000000000000002]

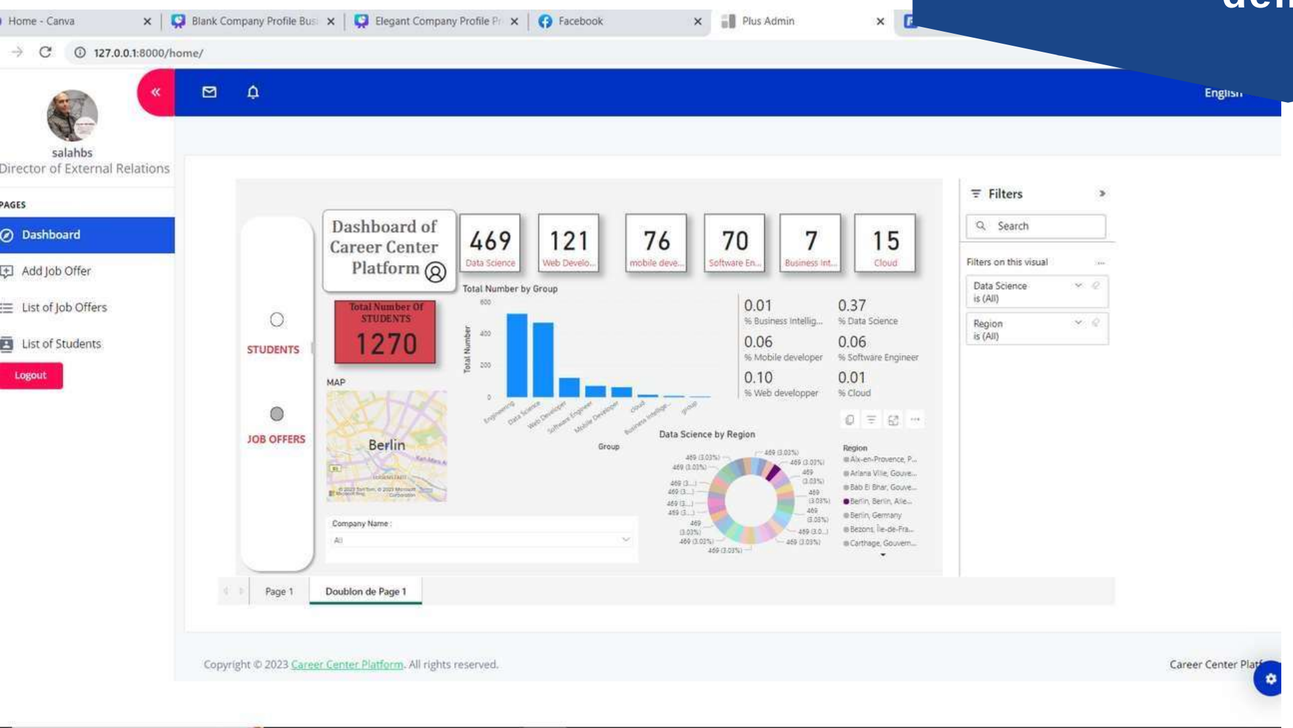
DEPLOYMENT

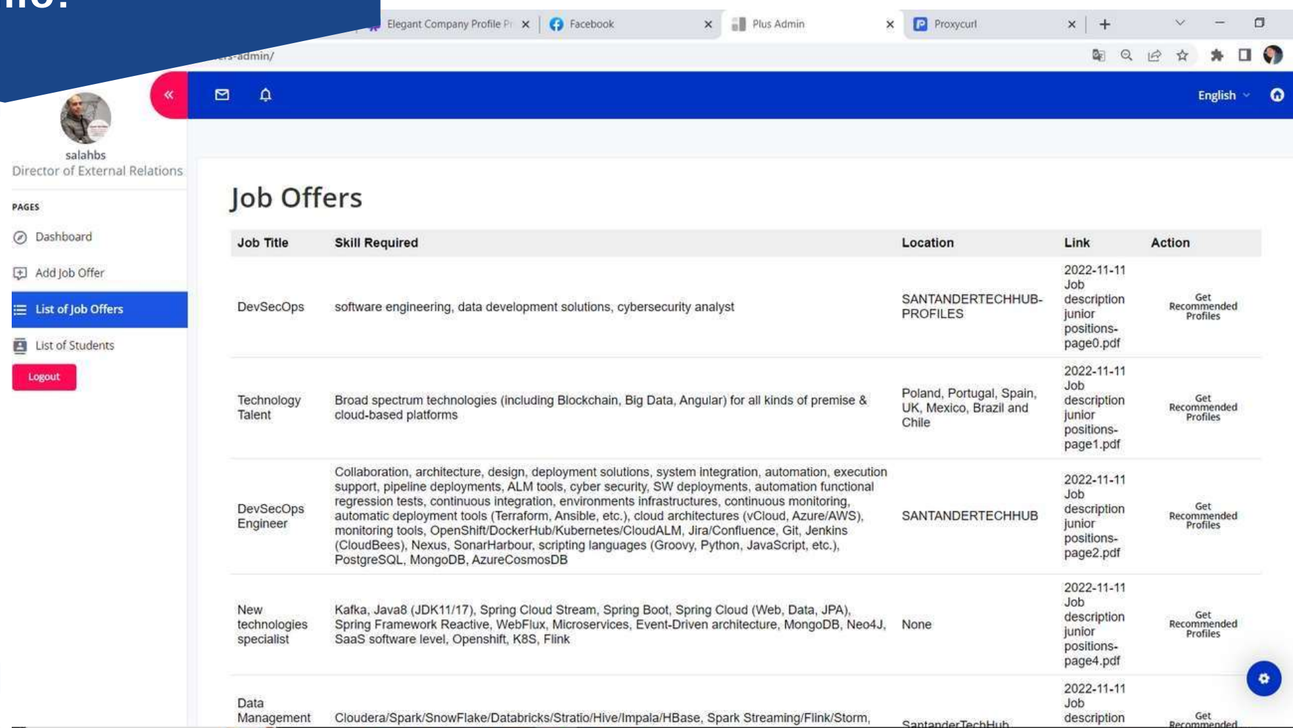
06





See for yourself the remarkable features in our demo!





THANK YOU FOR YOUR ATTENTION