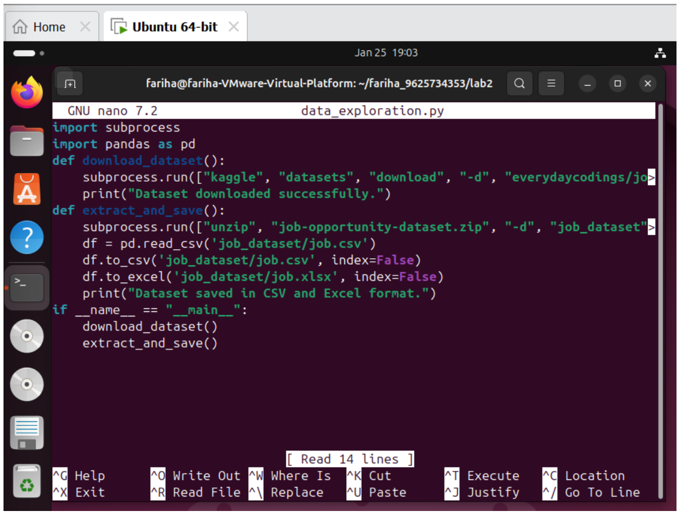
Data Science Professional Practicum (DSCI 560)

Laboratory Assignment 2

1. Team Formation
    a. Team name: DS Squad
    b. Team numbers and their USC IDs:
        i. Hanlu Ma (USC ID: 1392-9443-71)
        ii. Zhenyu Chen (USC ID: 2242-3773-15)
        iii. Fariha Sheikh (USC ID: 9625-7343-53)
    c. Lab 2 Repo Link: https://github.com/YiwenC23/DSCI560-group_lab2
2. Data selection, Search, Find, and Collect
    a. Domain: DS job seeking and preparation content
    b. Reason: The pandemic has dramatically impacted the economy, and many people have lost their jobs, especially those who work in technology companies. Even though the economy has been recovering in recent years, it is still difficult for people to find employment. Therefore, we decide to focus on the data science job-seeking and preparation domain, helping individuals find all currently open job positions that match their resumes, as well as assisting them to better prepare for their interviews based on the job descriptions.
    c. Dataset links and descriptions
        i. ASCII Texts like Forum Postings and HTML
            1. Link: https://news.ycombinator.com/item?id=24460141
            2. Description: this webpage entails peoples' responses about how to prepare for data scientist interviews. After the data exploration phase, this dataset will provide some 'empirical' and 'humane' advice and suggestions about data scientist interview preparation.
        ii. PDF and Word Documents that require conversion and OCR
            1. Link: https://www.kaggle.com/discussions/general/177093
            2. Description: the pdf file is obtained from the Kaggle website. It contains a comprehensive collection of Data Science interview questions that candidates may encounter during their interviews in the future, helping them better prepare.
        iii. CSV or Excel
            1. Link: www.kaggle.com/datasets/everydaycodings/job-opportunity-dataset
            2. The "Job Opportunities Dataset" is a comprehensive collection of information related to various job opportunities across diverse industries. This dataset provides details about the job title, company name, location, start date, CTC (Cost to Company), experience requirements, and the posted time for each job listing. The purpose of this dataset is to offer insights into the wide array of job opportunities available, catering to different roles and skill sets.
            3. This dataset helps my project in many ways. The "Job Title" helps the chatbot match positions based on users' skills and interests, while the "Company Name" allows users to filter jobs by specific employers. The "Location" column enables the chatbot to recommend jobs based on

geographic preferences, including options for work-from-home roles. "Start Date" ensures users can apply to positions with the most relevant or urgent start times. "CTC" (Cost to Company) allows salary filtering, helping users find jobs within their desired compensation range. The "Experience" column helps the chatbot suggest roles suited to the user's experience level, and "Posted" ensures users see the most recent job listings. Lastly, the dataset's job title and trend insights (such as the popularity of Business Development Executive roles) guide the chatbot in recommending high-demand positions. Together, these columns enable the chatbot to provide personalized job recommendations, making the job search more tailored, efficient, and user-friendly.

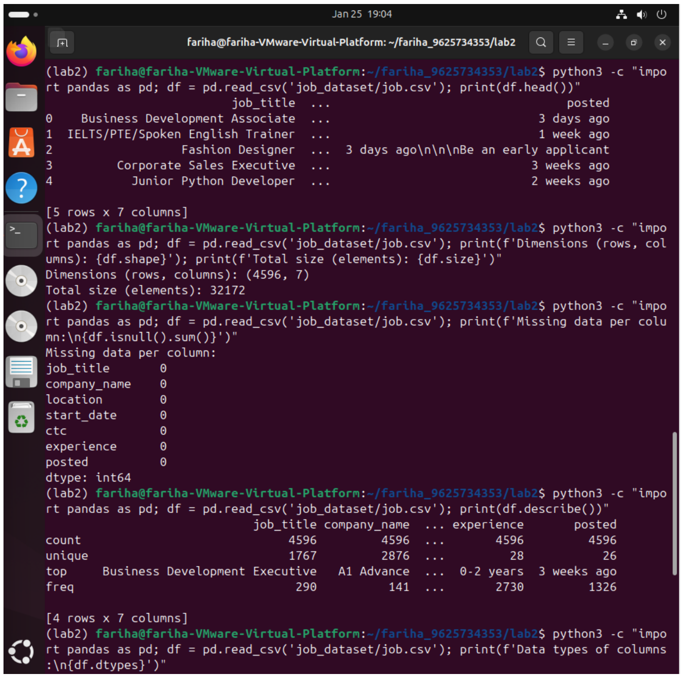
3. Data collection
    a. CSV or Excel
        i. Code



Home
Ubuntu 64-bit
Jan 25 19:03
fariha@fariha-VMware-Virtual-Platform: ~/fariha_9625734353/lab2

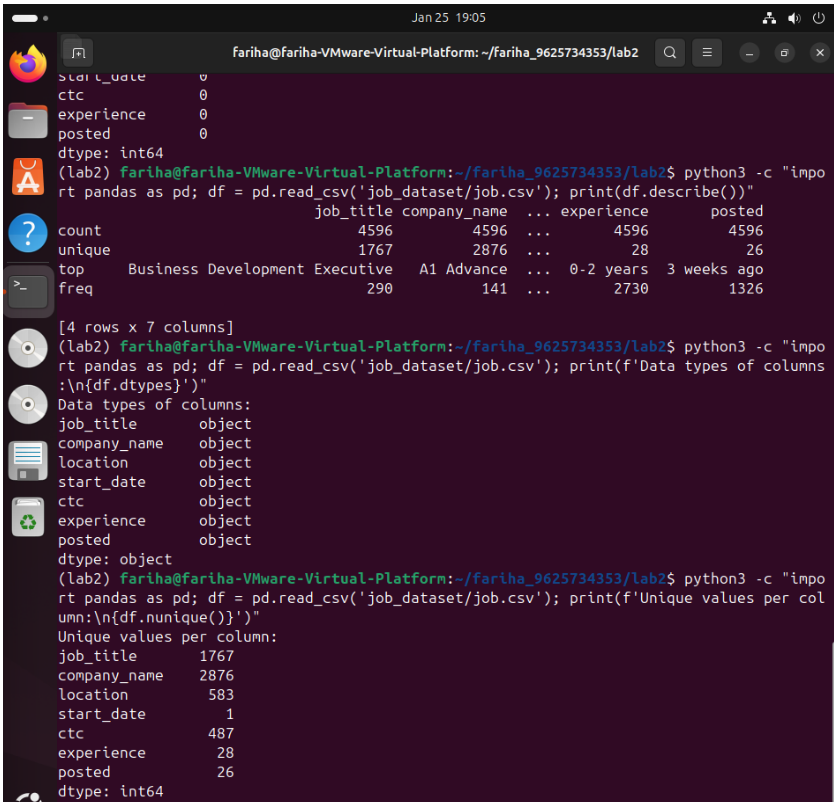
```
  GNU nano 7.2                        data_exploration.py
import subprocess
import pandas as pd
def download_dataset():
    subprocess.run(["kaggle", "datasets", "download", "-d", "everydaycodings/jo>
    print("Dataset downloaded successfully.")
def extract_and_save():
    subprocess.run(["unzip", "job-opportunity-dataset.zip", "-d", "job_dataset">
    df = pd.read_csv('job_dataset/job.csv')
    df.to_csv('job_dataset/job.csv', index=False)
    df.to_excel('job_dataset/job.xlsx', index=False)
    print("Dataset saved in CSV and Excel format.")
if __name__ == "__main__":
    download_dataset()
    extract_and_save()

                              [ Read 14 lines ]
^G Help       ^O Write Out ^W Where Is  ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File ^\ Replace   ^U Paste      ^J Justify    ^/ Go To Line
```

        ii. Output



Jan 25 19:04
fariha@fariha-VMware-Virtual-Platform: ~/fariha_9625734353/lab2

```
(lab2) fariha@fariha-VMware-Virtual-Platform:~/fariha_9625734353/lab2$ python3 -c "impo
rt pandas as pd; df = pd.read_csv('job_dataset/job.csv'); print(df.head())"
                          job_title  ...                                 posted
0    Business Development Associate  ...                             3 days ago
1  IELTS/PTE/Spoken English Trainer  ...                             1 week ago
2                  Fashion Designer  ...  3 days ago\n\n\nBe an early applicant
3         Corporate Sales Executive  ...                            3 weeks ago
4           Junior Python Developer  ...                            2 weeks ago

[5 rows x 7 columns]
(lab2) fariha@fariha-VMware-Virtual-Platform:~/fariha_9625734353/lab2$ python3 -c "impo
rt pandas as pd; df = pd.read_csv('job_dataset/job.csv'); print(f'Dimensions (rows, col
umns): {df.shape}'); print(f'Total size (elements): {df.size}')"
Dimensions (rows, columns): (4596, 7)
Total size (elements): 32172
(lab2) fariha@fariha-VMware-Virtual-Platform:~/fariha_9625734353/lab2$ python3 -c "impo
rt pandas as pd; df = pd.read_csv('job_dataset/job.csv'); print(f'Missing data per colu
mn:\n{df.isnull().sum()}')"
Missing data per column:
job_title       0
company_name    0
location        0
start_date      0
ctc             0
experience      0
posted          0
dtype: int64
(lab2) fariha@fariha-VMware-Virtual-Platform:~/fariha_9625734353/lab2$ python3 -c "impo
rt pandas as pd; df = pd.read_csv('job_dataset/job.csv'); print(df.describe())"
                             job_title company_name  ... experience       posted
count                             4596         4596  ...       4596         4596
unique                            1767         2876  ...         28           26
top     Business Development Executive   A1 Advance  ...  0-2 years  3 weeks ago
freq                               290          141  ...       2730         1326

[4 rows x 7 columns]
(lab2) fariha@fariha-VMware-Virtual-Platform:~/fariha_9625734353/lab2$ python3 -c "impo
rt pandas as pd; df = pd.read_csv('job_dataset/job.csv'); print(f'Data types of columns
:\n{df.dtypes}')"
```



Jan 25 19:05
fariha@fariha-VMware-Virtual-Platform: ~/fariha_9625734353/lab2

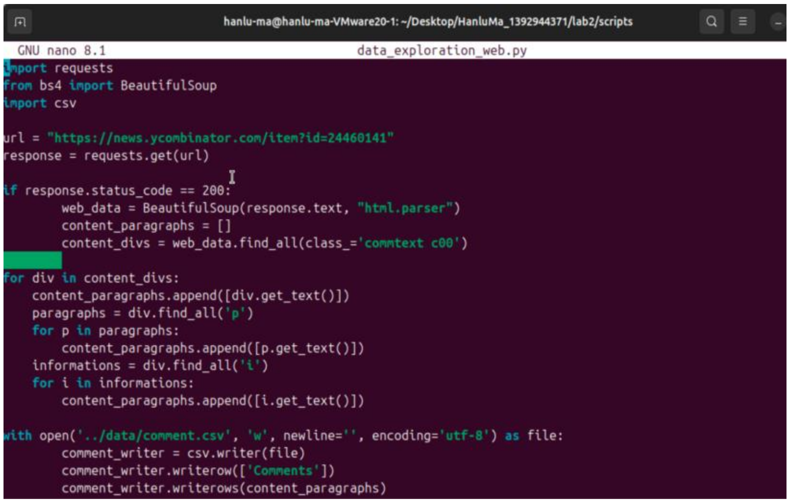
```
start_date      0
ctc             0
experience      0
posted          0
dtype: int64
(lab2) fariha@fariha-VMware-Virtual-Platform:~/fariha_9625734353/lab2$ python3 -c "impo
rt pandas as pd; df = pd.read_csv('job_dataset/job.csv'); print(df.describe())"
                             job_title company_name  ... experience       posted
count                             4596         4596  ...       4596         4596
unique                            1767         2876  ...         28           26
top     Business Development Executive   A1 Advance  ...  0-2 years  3 weeks ago
freq                               290          141  ...       2730         1326

[4 rows x 7 columns]
(lab2) fariha@fariha-VMware-Virtual-Platform:~/fariha_9625734353/lab2$ python3 -c "impo
rt pandas as pd; df = pd.read_csv('job_dataset/job.csv'); print(f'Data types of columns
:\n{df.dtypes}')"
Data types of columns:
job_title       object
company_name    object
location        object
start_date      object
ctc             object
experience      object
posted          object
dtype: object
(lab2) fariha@fariha-VMware-Virtual-Platform:~/fariha_9625734353/lab2$ python3 -c "impo
rt pandas as pd; df = pd.read_csv('job_dataset/job.csv'); print(f'Unique values per col
umn:\n{df.nunique()}')"
Unique values per column:
job_title       1767
company_name    2876
location         583
start_date         1
ctc              487
experience        28
posted            26
dtype: int64
```

iii. Rationale behind code

The data_exploration.py script downloads the "Job Opportunity Dataset" from Kaggle using the Kaggle API, unzips the downloaded file, and saves the dataset in both CSV and Excel formats. After extracting the data, it performs several basic data exploration tasks: it displays the first few records of the dataset, calculates and prints the size and dimensions (rows and columns), identifies any missing data, and provides a summary of basic statistics such as mean, standard deviation, etc. Additionally, it checks the data types of each column and counts the unique values in each column, offering a comprehensive overview of the dataset's structure and content.

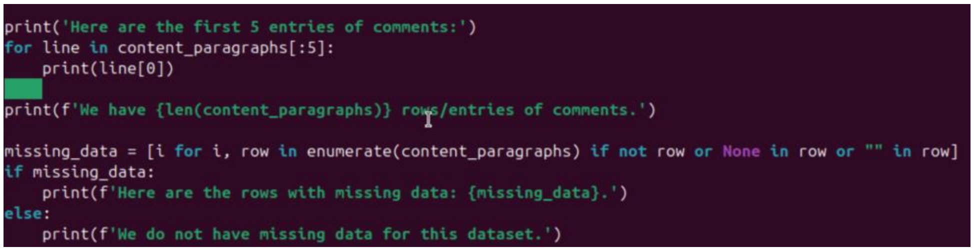
b. ASCII Texts like Forum Postings and HTML

i. Code



hanlu-ma@hanlu-ma-VMware20-1: ~/Desktop/HanluMa_1392944371/lab2/scripts
GNU nano 8.1
data_exploration_web.py

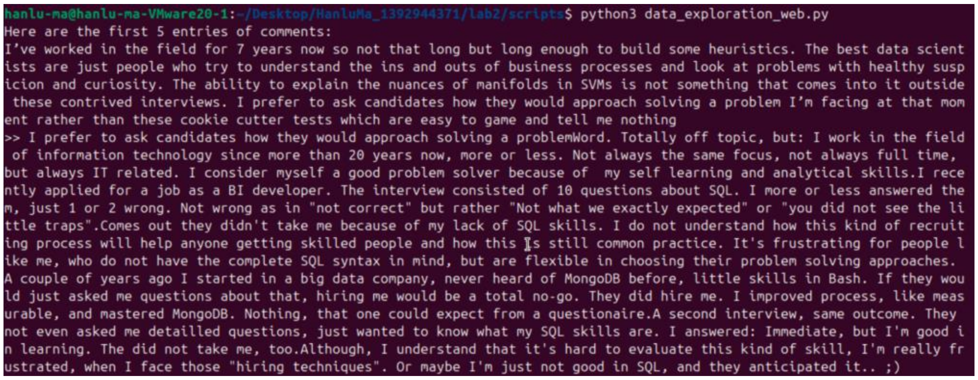
```
import requests
from bs4 import BeautifulSoup
import csv

url = "https://news.ycombinator.com/item?id=24460141"
response = requests.get(url)

if response.status_code == 200:
        web_data = BeautifulSoup(response.text, "html.parser")
        content_paragraphs = []
        content_divs = web_data.find_all(class_='commtext c00')

for div in content_divs:
    content_paragraphs.append([div.get_text()])
    paragraphs = div.find_all('p')
    for p in paragraphs:
        content_paragraphs.append([p.get_text()])
    informations = div.find_all('i')
    for i in informations:
        content_paragraphs.append([i.get_text()])

with open('../data/comment.csv', 'w', newline='', encoding='utf-8') as file:
        comment_writer = csv.writer(file)
        comment_writer.writerow(['Comments'])
        comment_writer.writerows(content_paragraphs)
```



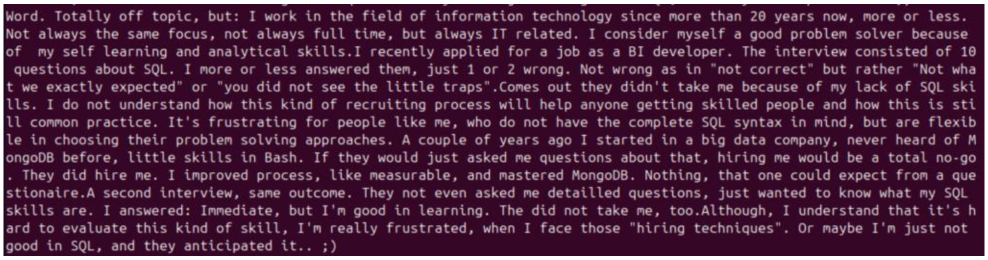
```
print('Here are the first 5 entries of comments:')
for line in content_paragraphs[:5]:
    print(line[0])

print(f'We have {len(content_paragraphs)} rows/entries of comments.')

missing_data = [i for i, row in enumerate(content_paragraphs) if not row or None in row or "" in row]
if missing_data:
    print(f'Here are the rows with missing data: {missing_data}.')
else:
    print(f'We do not have missing data for this dataset.')
```

ii. Output

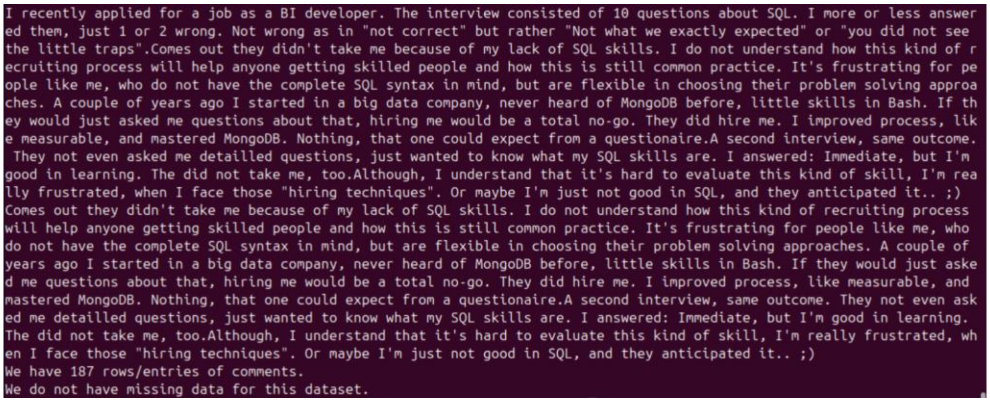
For data exploration, we generate the first 5 entries of comments, the number of comments/rows/entries, and the missing data situation to check the validity of the dataset.



```
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/lab2/scripts$ python3 data_exploration_web.py
Here are the first 5 entries of comments:
I’ve worked in the field for 7 years now so not that long but long enough to build some heuristics. The best data scient
ists are just people who try to understand the ins and outs of business processes and look at problems with healthy susp
icion and curiosity. The ability to explain the nuances of manifolds in SVMs is not something that comes into it outside
 these contrived interviews. I prefer to ask candidates how they would approach solving a problem I’m facing at that mom
ent rather than these cookie cutter tests which are easy to game and tell me nothing
>> I prefer to ask candidates how they would approach solving a problemWord. Totally off topic, but: I work in the field
 of information technology since more than 20 years now, more or less. Not always the same focus, not always full time,
but always IT related. I consider myself a good problem solver because of  my self learning and analytical skills.I rece
ntly applied for a job as a BI developer. The interview consisted of 10 questions about SQL. I more or less answered the
m, just 1 or 2 wrong. Not wrong as in "not correct" but rather "Not what we exactly expected" or "you did not see the li
ttle traps".Comes out they didn't take me because of my lack of SQL skills. I do not understand how this kind of recruit
ing process will help anyone getting skilled people and how this is still common practice. It's frustrating for people l
ike me, who do not have the complete SQL syntax in mind, but are flexible in choosing their problem solving approaches.
A couple of years ago I started in a big data company, never heard of MongoDB before, little skills in Bash. If they wou
ld just asked me questions about that, hiring me would be a total no-go. They did hire me. I improved process, like meas
urable, and mastered MongoDB. Nothing, that one could expect from a questionaire.A second interview, same outcome. They
not even asked me detailled questions, just wanted to know what my SQL skills are. I answered: Immediate, but I'm good i
n learning. The did not take me, too.Although, I understand that it's hard to evaluate this kind of skill, I'm really fr
ustrated, when I face those "hiring techniques". Or maybe I'm just not good in SQL, and they anticipated it.. ;)
```



```
Word. Totally off topic, but: I work in the field of information technology since more than 20 years now, more or less.
Not always the same focus, not always full time, but always IT related. I consider myself a good problem solver because
of  my self learning and analytical skills.I recently applied for a job as a BI developer. The interview consisted of 10
 questions about SQL. I more or less answered them, just 1 or 2 wrong. Not wrong as in "not correct" but rather "Not wha
t we exactly expected" or "you did not see the little traps".Comes out they didn't take me because of my lack of SQL ski
lls. I do not understand how this kind of recruiting process will help anyone getting skilled people and how this is sti
ll common practice. It's frustrating for people like me, who do not have the complete SQL syntax in mind, but are flexib
le in choosing their problem solving approaches. A couple of years ago I started in a big data company, never heard of M
ongoDB before, little skills in Bash. If they would just asked me questions about that, hiring me would be a total no-go
. They did hire me. I improved process, like measurable, and mastered MongoDB. Nothing, that one could expect from a que
stionaire.A second interview, same outcome. They not even asked me detailled questions, just wanted to know what my SQL
skills are. I answered: Immediate, but I'm good in learning. The did not take me, too.Although, I understand that it's h
ard to evaluate this kind of skill, I'm really frustrated, when I face those "hiring techniques". Or maybe I'm just not
good in SQL, and they anticipated it.. ;)
```



```
I recently applied for a job as a BI developer. The interview consisted of 10 questions about SQL. I more or less answer
ed them, just 1 or 2 wrong. Not wrong as in "not correct" but rather "Not what we exactly expected" or "you did not see
the little traps".Comes out they didn't take me because of my lack of SQL skills. I do not understand how this kind of r
ecruiting process will help anyone getting skilled people and how this is still common practice. It's frustrating for pe
ople like me, who do not have the complete SQL syntax in mind, but are flexible in choosing their problem solving approa
ches. A couple of years ago I started in a big data company, never heard of MongoDB before, little skills in Bash. If th
ey would just asked me questions about that, hiring me would be a total no-go. They did hire me. I improved process, lik
e measurable, and mastered MongoDB. Nothing, that one could expect from a questionaire.A second interview, same outcome.
 They not even asked me detailled questions, just wanted to know what my SQL skills are. I answered: Immediate, but I'm
good in learning. The did not take me, too.Although, I understand that it's hard to evaluate this kind of skill, I'm rea
lly frustrated, when I face those "hiring techniques". Or maybe I'm just not good in SQL, and they anticipated it.. ;)
Comes out they didn't take me because of my lack of SQL skills. I do not understand how this kind of recruiting process
will help anyone getting skilled people and how this is still common practice. It's frustrating for people like me, who
do not have the complete SQL syntax in mind, but are flexible in choosing their problem solving approaches. A couple of
years ago I started in a big data company, never heard of MongoDB before, little skills in Bash. If they would just aske
d me questions about that, hiring me would be a total no-go. They did hire me. I improved process, like measurable, and
mastered MongoDB. Nothing, that one could expect from a questionaire.A second interview, same outcome. They not even ask
ed me detailled questions, just wanted to know what my SQL skills are. I answered: Immediate, but I'm good in learning.
The did not take me, too.Although, I understand that it's hard to evaluate this kind of skill, I'm really frustrated, wh
en I face those "hiring techniques". Or maybe I'm just not good in SQL, and they anticipated it.. ;)
We have 187 rows/entries of comments.
We do not have missing data for this dataset.
```

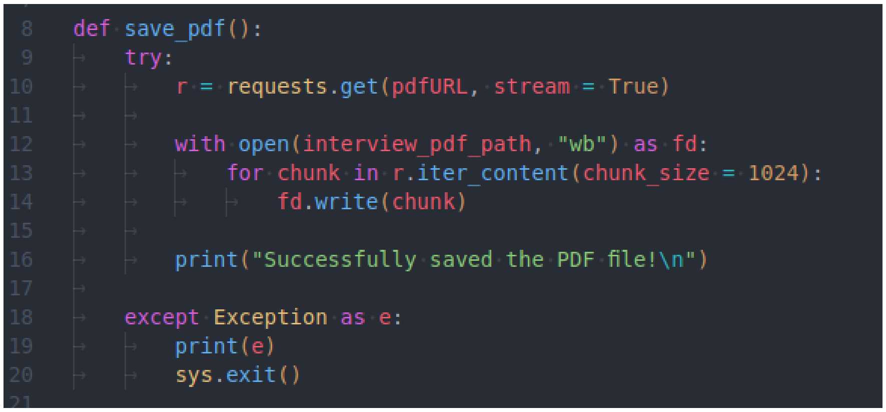
iii. Rationale of Code

1. We use the ‘get’ method from the requests library to get the link and utilize ‘BeautifulSoup’ to obtain the information. Then we locate the html location of the context we want to parse, which is ‘commtext c00’ in this case. After deep diving into the webpage html, we notice that some comments go along with the div, but others appear with ‘p’ or ‘i’ tag under the ‘commtext c00’ div. Therefore, we use the ‘final_all’ method and the ‘get_text’ method to both obtain information along with the div and within the category of the corresponding div. After obtaining the relevant information and storing in the relevant variable ‘content_paragraphs’, we write it into csv under the column name of

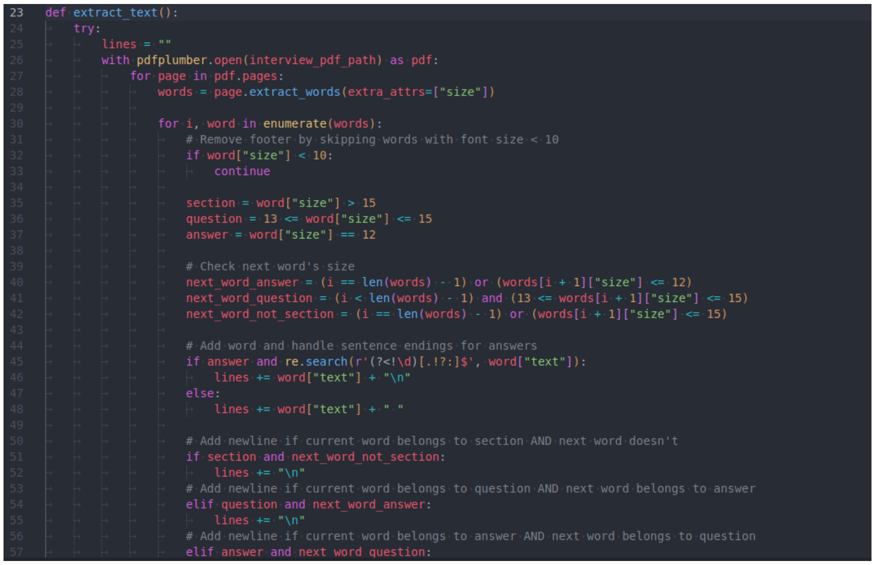
‘Comments’. Lastly, we have few lines to do some sanity checks about the dataset in terms of dataset length, missing values, and sample data extraction.

c. PDF and Word Documents that require conversion and OCR

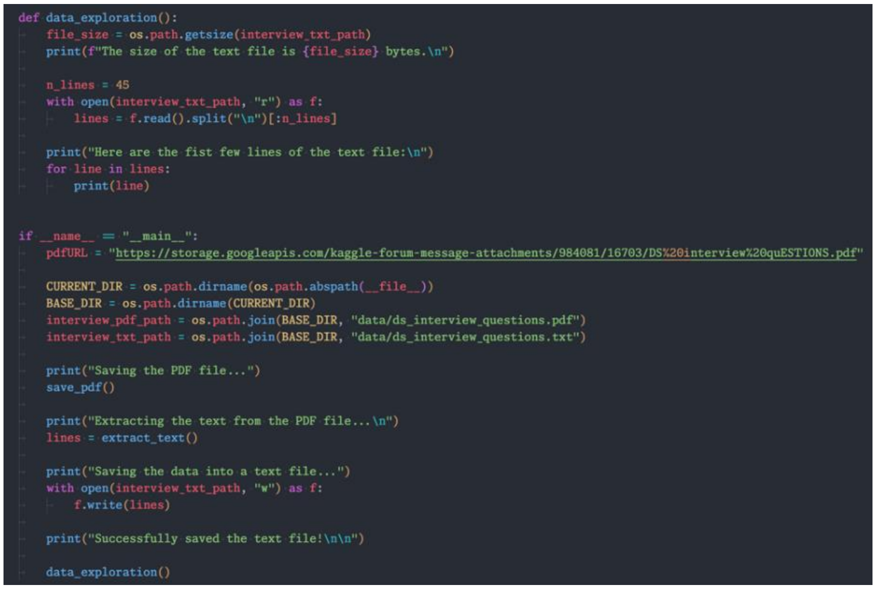
i. Code



```python
def save_pdf():
    try:
        r = requests.get(pdfURL, stream = True)

        with open(interview_pdf_path, "wb") as fd:
            for chunk in r.iter_content(chunk_size = 1024):
                fd.write(chunk)

        print("Successfully saved the PDF file!\n")

    except Exception as e:
        print(e)
        sys.exit()
```

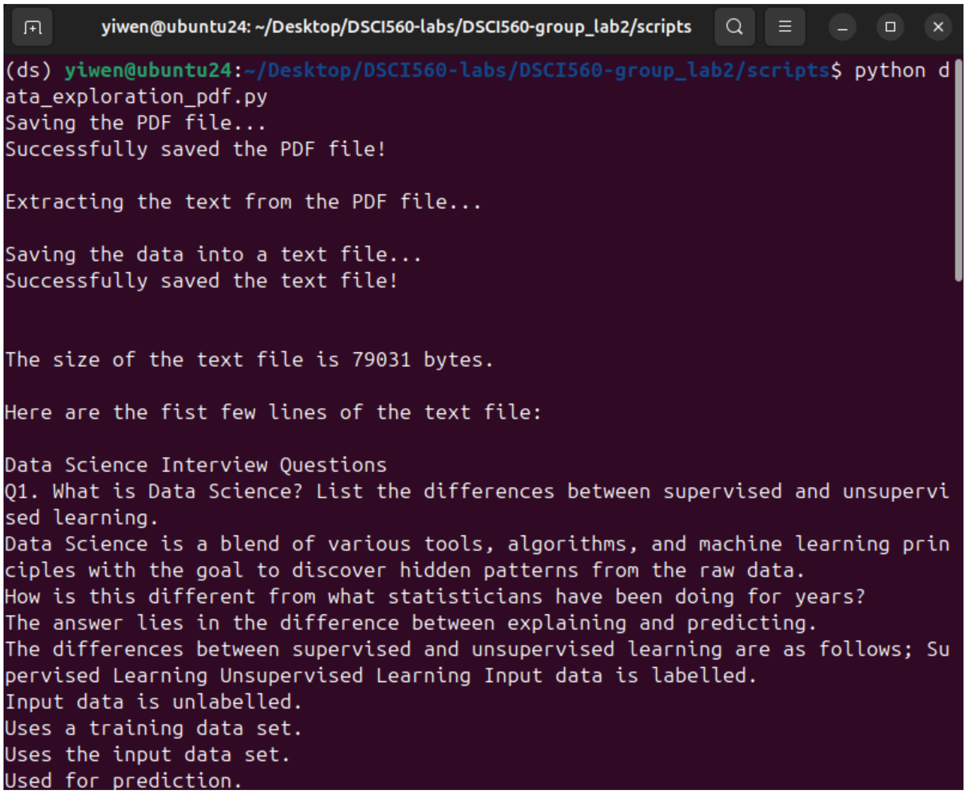


```python
def extract_text():
    try:
        lines = ""
        with pdfplumber.open(interview_pdf_path) as pdf:
            for page in pdf.pages:
                words = page.extract_words(extra_attrs=["size"])

                for i, word in enumerate(words):
                    # Remove footer by skipping words with font size < 10
                    if word["size"] < 10:
                        continue

                    section = word["size"] > 15
                    question = 13 <= word["size"] <= 15
                    answer = word["size"] == 12

                    # Check next word's size
                    next_word_answer = (i == len(words) - 1) or (words[i + 1]["size"] <= 12)
                    next_word_question = (i < len(words) - 1) and (13 <= words[i + 1]["size"] <= 15)
                    next_word_not_section = (i == len(words) - 1) or (words[i + 1]["size"] <= 15)

                    # Add word and handle sentence endings for answers
                    if answer and re.search(r'(?<!\d)[.!?:]$', word["text"]):
                        lines += word["text"] + "\n"
                    else:
                        lines += word["text"] + " "

                    # Add newline if current word belongs to section AND next word doesn't
                    if section and next_word_not_section:
                        lines += "\n"
                    # Add newline if current word belongs to question AND next word belongs to answer
                    elif question and next_word_answer:
                        lines += "\n"
                    # Add newline if current word belongs to answer AND next word belongs to question
                    elif answer and next_word_question:
```



```python
def data_exploration():
    file_size = os.path.getsize(interview_txt_path)
    print(f"The size of the text file is {file_size} bytes.\n")

    n_lines = 45
    with open(interview_txt_path, "r") as f:
        lines = f.read().split("\n")[:n_lines]

    print("Here are the fist few lines of the text file:\n")
    for line in lines:
        print(line)


if __name__ == "__main__":
    pdfURL = "https://storage.googleapis.com/kaggle-forum-message-attachments/984081/16703/DS%20interview%20quESTIONS.pdf"

    CURRENT_DIR = os.path.dirname(os.path.abspath(__file__))
    BASE_DIR = os.path.dirname(CURRENT_DIR)
    interview_pdf_path = os.path.join(BASE_DIR, "data/ds_interview_questions.pdf")
    interview_txt_path = os.path.join(BASE_DIR, "data/ds_interview_questions.txt")

    print("Saving the PDF file...")
    save_pdf()

    print("Extracting the text from the PDF file...\n")
    lines = extract_text()

    print("Saving the data into a text file...")
    with open(interview_txt_path, "w") as f:
        f.write(lines)

    print("Successfully saved the text file!\n\n")

    data_exploration()
```

ii. Output


yiwen@ubuntu24: ~/Desktop/DSCI560-labs/DSCI560-group_lab2/scripts

```
(ds) yiwen@ubuntu24:~/Desktop/DSCI560-labs/DSCI560-group_lab2/scripts$ python d
ata_exploration_pdf.py
Saving the PDF file...
Successfully saved the PDF file!

Extracting the text from the PDF file...

Saving the data into a text file...
Successfully saved the text file!


The size of the text file is 79031 bytes.

Here are the fist few lines of the text file:

Data Science Interview Questions
Q1. What is Data Science? List the differences between supervised and unsupervi
sed learning.
Data Science is a blend of various tools, algorithms, and machine learning prin
ciples with the goal to discover hidden patterns from the raw data.
How is this different from what statisticians have been doing for years?
The answer lies in the difference between explaining and predicting.
The differences between supervised and unsupervised learning are as follows; Su
pervised Learning Unsupervised Learning Input data is labelled.
Input data is unlabelled.
Uses a training data set.
Uses the input data set.
Used for prediction.
```

iii. Rationale of Code

First, we use the requests library to save the PDF file locally from the Kaggle webpage. Then, we use the "extract_words(extra_attrs=['fontname', 'size'])" method from the pdfplumber library to extract words from the PDF pages along with extra information – the font name and size of each word. After analyzing the font properties, we found that the font size of the section titles is 18, the font size of the question titles is 13, and the answer texts are size 12. Therefore, we distinguish the section lines, question title lines, and answers by comparing the current word's font size and the next word's font size. For instance, if the current word's font size is greater than 15 and the next font size is between 13 and 15, then we start a new line between these two words. With this method, we are able to organize the content into three different lines. Moreover, for the answers' line breaking, we apply the re library to check if the word is followed by common punctuation symbols (",", "?", "!", and ":"). After we have processed the entire content in this format, we save the data into a text file locally.