DEPARTMENT OF COMPUTING AND INFORMATION SYSTEMS
SCHOOL OF ENGINEERING AND TECHNOLOGY

FINAL ASSESSMENT FOR:
BSC (HONS) INFORMATION SYSTEMS (DATA ANALYTICS);
BACHELOR OF SCIENCE (HONOURS) IN COMPUTER SCIENCE (BCS)

ACADEMIC SESSION: APRIL 2024

SWA2124: SOCIAL AND WEB ANALYTICS

DEADLINE: 14th June 2024 (Friday), by 11:59pm.

GROUP NAME : InsideOut

INSTRUCTIONS TO CANDIDATES

- This assignment will contribute **25%** to your final grade.
- This project is a **GROUP** assignment. Each group consists of 3 - 4 members.

IMPORTANT

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

Student's Declaration:

	(Name)	(ID)	(Signature)
We	1) Chia Wan Ying	23020829	*wanying*
	2) Tai Yong Xuan	22012835	*yongxuan*
	3) Ooi Shi Qi	21098272	*shiqi*
	4) Terrence Teoh Jin Haw	23027311	*terrence*

received the assignment and read the comments.

Academic Honesty Acknowledgement

"We (names stated above) verify that this paper contains entirely my own work. I have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, I have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. I realize the penalties *(refer to the student handbook)* for any kind of copying or collaboration on any assignment."

1)Chia Wan Ying

2)Tai Yong Xuan

3)Ooi Shi Qi

4)Terrence Teoh Jin Haw

Wanying yongxuan shiqi Terrence 21/6/2024

……

(Student's Signature and Date)

1.0 Introduction

In this assignment, we will be using JobStreet as our choice of job vacancy website to perform our analysis. As for our reason for our choice of website, it is because JobStreet provides detailed reviews from past employees including the benefits, challenges and ratings. It also provides a detailed description of the company with a decent amount of sample size. Ratings are as well detailed as it is categorized into work and life balance, career development, benefits and perks, management, working environment and, diversity and equal opportunity. These detailed reviews enable us to perform analysis and research to generate deep insights and valuable information.

The company of choice for our analysis is Top Glove as it is a well-known local business enterprise in Malaysia that is striving in the manufacturing, transport and logistics industries. With its company size having more than 10,000 employees, it comes with more than 350 sample reviews that will be sufficient for our research and analysis. Top Glove Corporation Bhd is a Malaysian company that is the world's largest producer of rubber, nitrile, and surgical gloves. Tan Sri Dr. Lim Wee Chai founded the company in 1991, and it has since expanded significantly, exporting to over 195 countries. Top Glove's success stems from its extensive production capacity, technological advancements, and strategic acquisitions.

In this analysis, we will be using different modules such as the Selenium WebDriver module and the Natural Language Toolkit (NLKT) python package. The purpose of using the Selenium WebDriver module is to interact with the web browsers and locate the elements that we want to scrape while the NLKT python package serves as a text preprocessing tool to analyze text (Krukowski, 2024). The main difference from what we have cited is that we used Cascading Style Sheets to locate the elements that we want to scrape. In this case, it would be the reviews and we would identify its elements on the website.

2.0 Coding in python

2.1 Importing constants, functions, and modules

```
# Importing constants, functions, and modules
from csv import QUOTE_ALL, QUOTE_NONNUMERIC
from time import sleep
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from selenium.common.exceptions import NoSuchElementException
from selenium.webdriver.common.action_chains import ActionChains
import os
import pandas as pd
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.probability import FreqDist
from nltk.sentiment import SentimentIntensityAnalyzer
from nltk import pos_tag
from collections import Counter
import os
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
```

Importing CSV module helps to handle CSV quoting styles. The ‘sleep’ in time module is used to introduce delays in program execution. Importing Selenium WebDriver modules such as ‘webdriver’ are used to execute cross-browser tests, ‘WebDriverWait’ to ensure elements are fully loaded, ‘By’ to locate elements in WebDriver, ‘NoSuchElementException’ to handle missing elements, and ‘ActionChains’ to perform actions. Standard Library imports includes ‘os’, the operating system, ‘pandas’ to work with data sets, ‘numpy’ is used for numerical computations, ‘matplotlib.pyploy’ for plotting graphs and charts, and ‘matplotlib.style’ to customize the style of the plot. Lastly, importing NLTK,natural language toolkit, can build Python programs to work with human language. ‘word_tokenize’, ‘stopwords’, ‘FreqDist’, ‘SentimentIntensityAnalyzer’, ‘pos_tag’, and ‘counter’ are used to process text and analyze it. Matplotlib is imported to produce plots to visualize data such as bar charts, line chart, pie chart etc. Importing seaborn allows the program to provide high-level interface for statistical graphics such as swarm chart. Lastly, re in Python which is regular expression is used to work

with regular expressions in Python. It helps to search patterns and extract information based on the criteria given.

2.2 Scrape ratings

```python
# Scrape ratings of Top Glove from JobStreet using selenium webdriver
def get_ratings(
    driver: webdriver.Chrome | webdriver.Firefox | webdriver.Safari | webdriver.Edge,
    url: str,
):
    driver.get(url)

    css_selector = "div[id^='review-card-'] > :nth-child(1) > :nth-child(1) > :nth-child(1) >
    page = 0
    ratings: list[float] = []

    while len(ratings) < 200:
        page += 1
        WebDriverWait(driver, 10).until(
            EC.presence_of_all_elements_located((By.CSS_SELECTOR, css_selector))
        )
        elems = driver.find_elements(By.CSS_SELECTOR, css_selector)
        ratings_in_page = [float(e.text) for e in elems]
        print(page, ratings_in_page)
        if ratings_in_page:
            ratings.extend(ratings_in_page)
        try:
            next_button = driver.find_element(
                By.CSS_SELECTOR, "a[title='Next'][aria-hidden='false']"
            )
            if next_button:
                next_button.click()
                sleep(1)
        except NoSuchElementException:
            break
    return ratings[:200]
```

def get_ratings() function is used to scrape ratings of Top Glove from the website jobstreet.com.my using Selenium WebDriver. The purpose of this function is to navigate to the provided URL, extract rating from it, click through to the next page and collect a list of ratings until there is no next page available. The ‘driver’ is used to control web browsers such as Chrome, Firefox, Safari, and Edge to interact with its web elements. Css_selector is a simple way used to locate the elements on the webpage. While len(ratings)<200 ensures that number of data scraped is 200. Try and except is used to handle pagination where the WebDriver presses the ‘Next’ button until no button is found.

2.3 Scrape reviews

```python
# Scrape reviews of Top Glove from Jobstreet using selenium webdriver
def get_reviews(
    driver: webdriver.Chrome | webdriver.Firefox | webdriver.Safari | webdriver.Edge,
    url: str,
):
    limit = 200
    driver.get(url)
    company_name_selector = "div#app > div > div > div > :nth-child(1) > :nth-child(2) >
    position_selector = ":nth-child(1) > :nth-child(1) > :nth-child(1) > :nth-child(1) >
    summary_selector = "h4"
    rating_selector = "div > div > div > div > div > div > div > div:nth-child(1) > div
    good_selector = "div > div > div > div > div:nth-child(2)> div > div > div:nth-child
    challenges_selector = "div > div > div > div > div:nth-child(2) > div > div > div:nt
    reviews = []
    page = 0
    chains = ActionChains(driver)

    company_name = driver.find_element(
        By.CSS_SELECTOR, company_name_selector).text
    print(company_name)
```

```python
while len(reviews) < 200:
    page += 1
    WebDriverWait(driver, 10).until(
        EC.presence_of_all_elements_located((By.CSS_SELECTOR, "div[id^='review-card-']"))
    )

    review_cards = driver.find_elements(
        By.CSS_SELECTOR, "div[id^='review-card-']")
    for card in review_cards:
        rating = card.find_element(By.CSS_SELECTOR, rating_selector)
        position = card.find_element(By.CSS_SELECTOR, position_selector)
        summary = card.find_element(By.CSS_SELECTOR, summary_selector)
        good = card.find_element(By.CSS_SELECTOR, good_selector)
        challenges = card.find_element(
            By.CSS_SELECTOR, challenges_selector)
        reviews.append(
            {
                "rating": float(rating.text),
                "position": position.text,
                "summary": summary.text,
                "good": good.text,
                "challenges": challenges.text
            }
        )
        if limit > 0 and len(reviews) == limit:
            break
    print(f"Page {page:3d} ✅")
```

```
    if limit > 0 and len(reviews) == limit:
        break

    try:
        next_button = driver.find_element(
            By.CSS_SELECTOR, "a[title='Next'][aria-hidden='false']"
        )
        if next_button:
            chains.scroll_to_element(next_button)
            next_button.click()
            sleep(1)
    except NoSuchElementException:
        break
return reviews[:200]
```

Def get_reviews() function is used to scrape reviews of Top Glove from jobstreet.com.my using selenium WebDriver. Company_name_selector finds and prints the company name from the page. An empty list reviews = [] is used to store the data scrapped from the webpage. ActionChains is used to scroll through the page and all the css_selectors are used to locate the elements' position.

The function loops through by finding the review-card on the page. Then it extracts the data based on the selectors such as rating, position, summary, good, and challenges. It then appends the scraped data into the list. When the number of extracted reviews reaches 200, the loop breaks. The code will look for 'Next' button on the page, clicks it then waits for half a second for it to load. If no such element is found, the loop breaks. Then it returns to the list.

2.4 Analyze text of reviews

```python
# Analyze text of reviews
def analyze_reviews(df: pd.DataFrame):

    reviews = df['good'].tolist() + df['challenges'].tolist()

    # Tokenize and preprocess the reviews
    stop_words = set(stopwords.words('english'))
    all_words = []
    for review in reviews:
        words = word_tokenize(review)
        words = [word.lower() for word in words if word.isalnum()
                 and word.lower() not in stop_words]
        all_words.extend(words)

    # Calculate word frequencies
    freq_dist = FreqDist(all_words)

    # Use SentimentIntensityAnalyzer to determine sentiment
    sia = SentimentIntensityAnalyzer()
    positive_words = []
    negative_words = []
    for word, frequency in freq_dist.items():
        sentiment_score = sia.polarity_scores(word)['compound']
        if sentiment_score >= 0.5:
            positive_words.append((word, frequency))
        elif sentiment_score <= -0.5:
            negative_words.append((word, frequency))

    # Sort words by frequency and print top 5
    positive_words.sort(key=lambda x: x[1], reverse=True)
    negative_words.sort(key=lambda x: x[1], reverse=True)

    print("Top 5 positive keywords:")
    for word, frequency in positive_words[:5]:
        print(f"{word}: {frequency}")

    print("\nTop 5 negative keywords:")
    for word, frequency in negative_words[:5]:
        print(f"{word}: {frequency}")

    return positive_words , negative_words
```

The def analyze_reviews(df: pd.DataFrame): function is used to analyze text of reviews in the DataFrame. In this function, it tokenize and preprocesses the reviews, calculate word frequencies, determine sentiment of each word, and sort words. It first combines the 'good' and 'challenges' column of the DataFrame into a list then it converts the words into lowercase, filters out non-alphanumeric words, and remove stopwords. Stopwords are words like 'a', 'the', 'are' etc. The tokenize words are stored in the list called all_words. It then calculates the frequencies of each word and analyze the sentiment of each word to append it to positive_words and negative_words list respectively based on the sentiment score of > 0.5 or <0.5. After appending it, it will sort and print to top 5 positive and negative words.

2.5 Preprocess list of words

```
#Preprocess list of words
def filtering(words):
  words = [word.lower() for word in words]
  tagged_words = pos_tag(words)
  filter_tags = {
      'CC', 'DT', 'EX', 'IN','LS',
      'MD','NNP','NNPS','PRP$','SYM',
      'TO','UH','WDT','WP','WP$','WRB'
  }
  filtered_words = [word for word, pos in tagged_words if pos not in filter_tags and word != 'i']
  return filtered_words
```

Def filtering(words) function is used to convert all the words to lowercase, tag the words, filter out words based on POS tags, and remove the word 'I'. This improves the accuracy by filtering out irrelevant words.

2.6 Analyze summary column of DataFrame

```python
# Analyze summary column of DataFrame
def count_repetitives(df: pd.DataFrame):
    column_name = "summary"
    words = df[column_name].str.lower().str.split()
    all_words = [word for sublist in words for word in sublist]
    all_words = filtering(all_words)
    word_counts = Counter(all_words)
    top_5_words = word_counts.most_common(5)

    print("The top 5 most repetitive words are:")
    for word, count in top_5_words:
        print(f"{word} : {count}")

    return words, top_5_words
```

Def count_repititives(df: pd.DataFrame) function is used to analyze the summary column of DataFrame and count the frequencies of the occurrence of each words after filtering out irrelevant words.

2.7 Main Function

```python
def main():
    url = "https://www.jobstreet.com.my/companies/top-glove-168556710434867/reviews"

    #download NLTK resources
    nltk.download('punkt', quiet=True)
    nltk.download('stopwords', quiet=True)
    nltk.download('vader_lexicon', quiet=True)
    nltk.download('averaged_perceptron_tagger', quiet=True)

    #configures headless Chrome WebDriver
    options = webdriver.ChromeOptions()
    binary_location = os.path.join(os.getcwd(), "chrome-linux64/chrome")
    options.binary_location = binary_location
    options.add_argument("--headless")
    options.add_argument("--no-sandbox")
    options.add_argument("--disable-dev-shm-usage")
    driver = webdriver.Chrome(options=options)

    try:
        result = get_reviews(driver, url)

        df = pd.DataFrame(result)
        df.to_csv("Top_Glove_Reviews.csv", quoting=QUOTE_NONNUMERIC)

        pw, nw = analyze_reviews(df)
        _, t = count_repetitives(df)
        calculate()
        review_line()
        summary_pie(t)
        challenges_bar_chart(nw)
        good_horizontal_barchart(pw)
        swarm()

    except Exception as e:
        print(e)
    finally:
        driver.close()
```

Def main() is the main function that scrape reviews from the given URL, process it, and analyze the contents then save it to a CSV file. Downloading NLTK resources are necessary to process, tokenize, analyze sentiment, and tag the text. Coding a headless Chrome browser allows the browser to run without a user interface. It is used for script running automation. Then, it will run the analyze_review(), count_repetitives(), calculate(), review_line(), summary_pie(), challenges_bar_chart(), good_horizontal_barchart(), and swarm() functions.

2.8 Plot a line chart for rating column

```
# Plot line chart for review column
def review_line():
    top_glove_r = pd.read_csv("Top_Glove_Reviews.csv")
    top_glove_r.head()  # Check the first few rows of the DataFrame

    ratings = np.array([1.0, 2.0, 3.0, 4.0, 5.0])
    ratings_count = top_glove_r['rating'].value_counts().reindex(
        ratings, fill_value=0)

    plt.plot(ratings,ratings_count.values, marker = '.')
    plt.xlabel('Rating Stars')
    plt.ylabel('Number of Ratings', fontsize=14)
    plt.title('Distribution of Ratings')

    plt.show()
```

The purpose of the review_line() function is to create a line chart of ratings column from CSV file. It defines the array for rating values and calculates the number of occurrences of each rating value. Then, it creates a plot with x-axis corresponding to y-axis with a marking of '.' At each point.

2.9 Calculate the mean

```
#Calculate the mean
def calculate():
    top_glove_rating = pd.read_csv("Top_Glove_Reviews.csv")
    mean_rating = np.mean(top_glove_rating['rating'])
    print(f"Mean rating: {mean_rating}\n")
```

The purpose of def calculate() function is to calculate the mean of the rating from Top_Glove_Reviews.csv by using numpy function.

2.10 Plot pie chart for summary column

```
#Plot pie chart for summary column
def summary_pie(top_5_words):
    labels = [x[0] for x in top_5_words]
    sizes = [x[1] for x in top_5_words]
    fig, ax = plt.subplots()
    ax.pie(sizes, labels=labels, autopct='%1.1f%%')
    plt.show()
```

The purpose of def summary_pie() function is to create and display a pie chart based on the top 5 words found in count_repititives() function. Top_5_words is a list of tuples so, list comprehension is needed to extract single element from the tuple. After extracting, it will generate a pie chart with added percentage label.

2.11 Plot swarm chart using Position and Ratings columns

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns
import re
```

This plot will be using the RegEx module, Seaborn module, Matplotlib library and the Pandas module. The pandas module will be interacting with the excel file while the RegEx module creates a search pattern and searches the data and categorizing them. As seaborn is based on the matplotlib library, it is used to create a visual plot data, in this case, we will be using it to create a swarm plot.

```
def swarm():
    # Load the CSV file
    df = pd.read_csv("Top_Glove_Reviews.csv")

    # Round ratings to the nearest whole number
    df['rating'] = df['rating'].round()

    # Drop rows with missing values in 'rating' or 'position'
    df = df.dropna(subset=['rating', 'position'])

    # Define categories based on regex patterns
    categories = {
        'Trainee': r'\b(trainee|intern|internship)\b',
        'Management': r'\b(manager|head|executive|supervisor|admin)\b',
        'Technical': r'\b(engineer|developer|technician|technical)\b',
    }

# Function to categorize positions based on regex patterns
def categorize_position(position):
    position_lower = position.lower()
    for category, pattern in categories.items():
        if re.search(pattern, position_lower, flags=re.IGNORECASE):
            return category

# Apply categorization function to create a new 'position_category' column
df['position_category'] = df['position'].apply(categorize_position)

# Create the swarm plot with switched x and y axes based on categories
plt.figure(figsize=(10, 6))
sns.set(style="whitegrid")

# Use hue and reduce marker size to handle overlapping points
swarm_plot = sns.swarmplot(x='position_category', y='rating', data=df, hue='position_category',\
             size=3, palette="viridis", dodge=True, legend=False)

# Customize the plot
plt.title('Ratings by Position Category at Top Glove', fontsize=16)
plt.xlabel('Position Category', fontsize=14)
plt.ylabel('Rating', fontsize=14)

# Round y-axis labels to whole numbers
plt.yticks(np.arange(df['rating'].min(), df['rating'].max() + 1, 1))

# Show the plot
plt.tight_layout()
plt.show()
```

The plot that will be coded is the swarm plot. The code starts off by reading the excel file that contains the reviews that were scraped previously. The next step is to ensure the data being at good quality. The code would also remove rows that does not have values for either the ratings or position value. The preprocessing process also includes classifying the data of the variable positions. The method used for classification is by using keywords to class data into 3 categories: Management, Technical and Trainee. The X-axis will be denoted as “Position Category” while the Y-axis will be denoted as “Rating”.

2.12 Good plot code

```
# Plot horizontal bar plot for top 5 positive keywords
def good_horizontal_barchart(positive_words):
    positive_w = [x[0] for x in positive_words][:5]
    f = [x[1] for x in positive_words][:5]
    plt.figure(figsize=(10, 6))
    plt.barh(positive_w, f, color='skyblue')
    plt.title('Top 5 Most Frequent Positive Keywords')
    plt.xlabel('Frequency')
    plt.ylabel('Positive Keywords')
    plt.grid(True)

    plt.show()
```

The purpose of this def good_horizontal_barchart() function is to create and display a horizontal bar chart from the top five positive words found from analyze_reviews function. List comprehension is used to extract a single element from the tuple. It then makes sure extract the first five positive words.

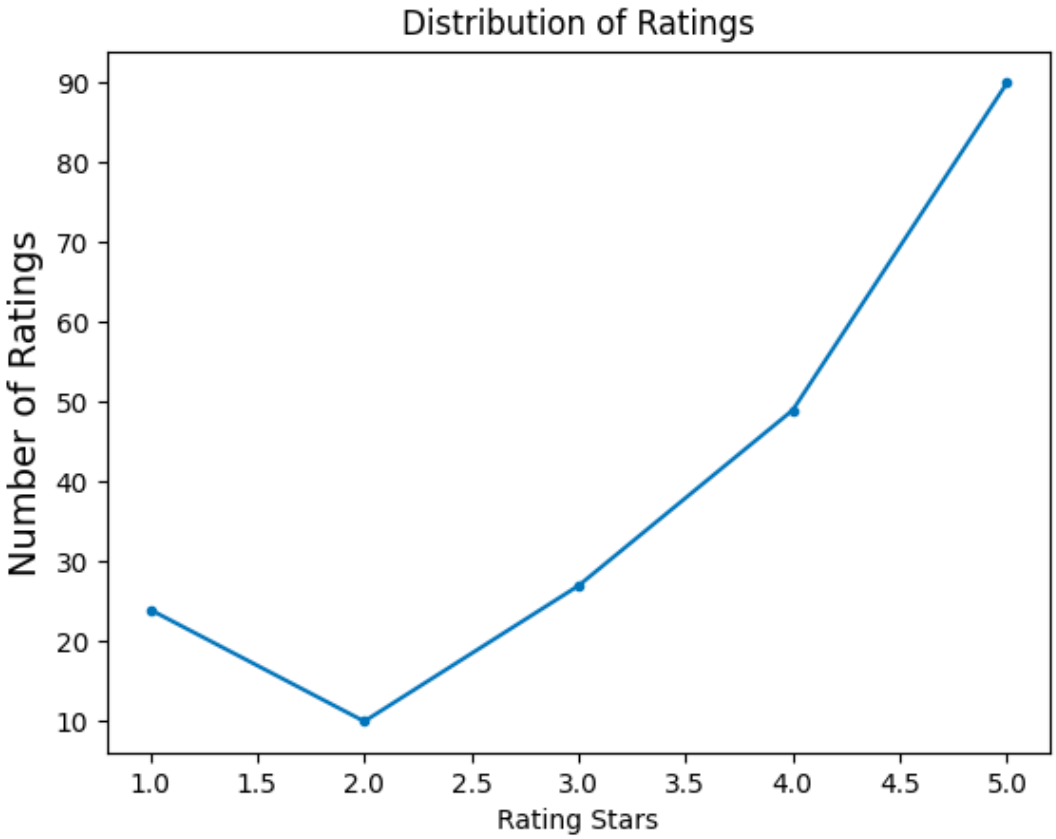
2.13 Plot bar chart for challenges column

```
#Plot bar chart for challenges column
def challenges_bar_chart(negative_words):
    negative_w = [x[0] for x in negative_words][:5]
    frequency = [x[1] for x in negative_words][:5]

    plt.figure(figsize=(10,5))
    plt.bar(negative_w, frequency,color = 'blue')
    plt.xlabel('Top 5 Negative Keywords')
    plt.ylabel('Frequency')
    plt.title('Frequency of Top 5 Negative Keywords in Reviews for Top Glove Graph')

    plt.show()
```

The purpose of def challenges_bar_chart() function is to create and display a bar chart based on the top 5 negative_words identified from analyze_reviews() function. List comprehension is used to extract a single element from the tuple.

3.0 Data Analysis

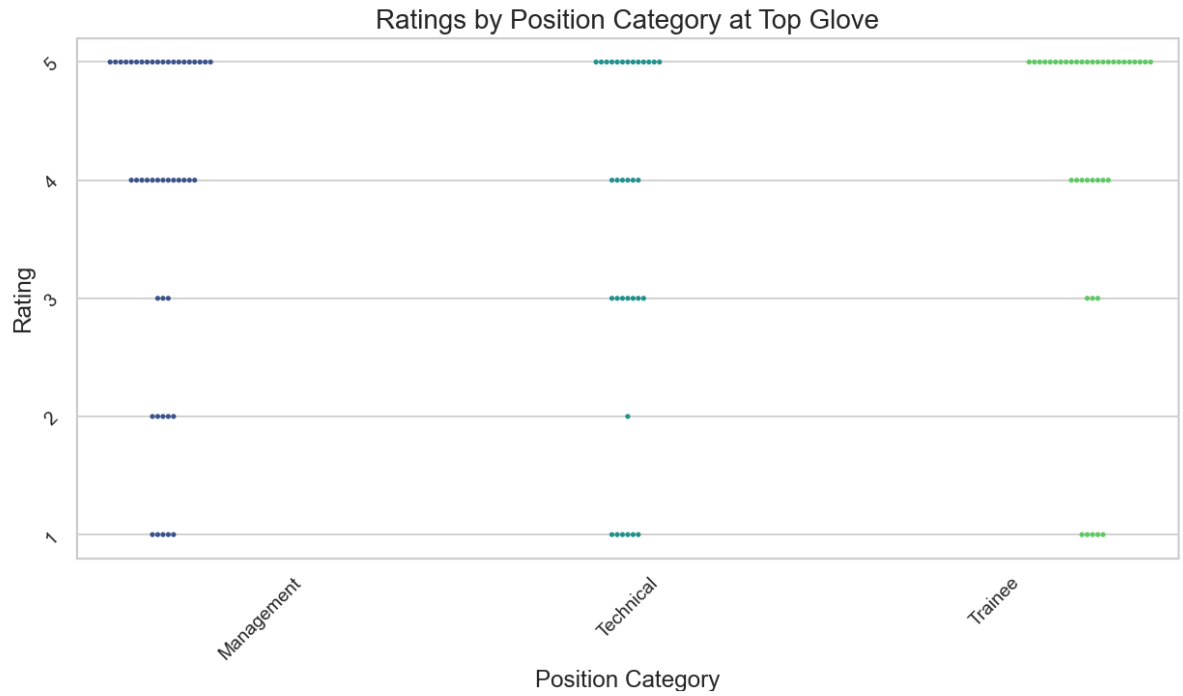
3.1 Rating



Analyzing the rating column is expected to find the number of ratings rated by employee of Top Glove by star rating. It provides insights to understand what employees likes and dislikes about the company in order to perform future improvements. The method used to analyze the star ratings is a line chart. Line chart display data in '.' Marker and connected by straight lines. The x-axis represents the star ratings and y-axis represents the number of occurrences in the first 200 reviews. Line chart is used to analyze star ratings as it is easy to compare on how many ratings are there in each star given. It also highlights the extreme points such as the lowest occurrence and the highest occurrence of star ratings.

In the first 200 reviews of Top Glove scraped from job street website, there are 25 reviews rated 1 star, 10 rated 2 stars, about 30 rated 3 stars, about 50 rated 4 stars, and about 90 rated 5 stars. It shows that 2.0 stars has the lowest count while 5.0 stars has the highest count. From def calculate() function, it shows that the mean of ratings are 3.82 stars which means Top Glove has meets the basic requirements. However, to identify it as good, it should be 4.0 to 4.5 stars and above. ("What makes a good star rating for products in different industries," n.d.).

3.2 Position and Ratings



This section aims to use a swarm plot that is coded in python to analyze the relationship between the ratings and position. The analysis expects to gain information on the level of satisfaction for each category of the position in Top Glove.

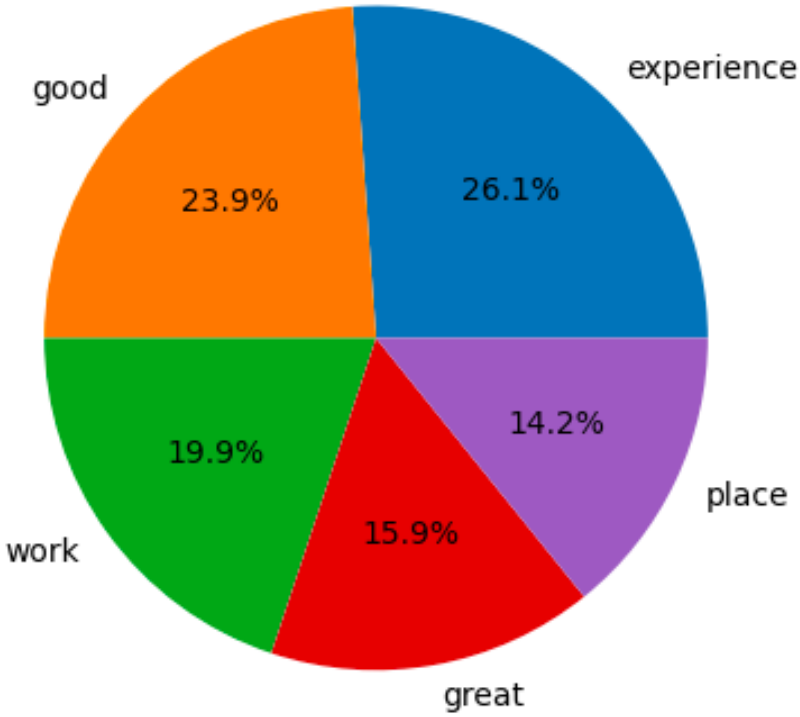
The positions will be classed into 3 categories: top management, technical sector and trainee. For the trainee category, they are mostly newcomers such as an intern or an individual that is currently undergoing training. Based on the swarm plot, it is shown that most of the trainees rated their position at Top Glove being 4 or 5 from the scale of 1 to 5. It is shown that it is highly recommended for individuals to have Top Glove as a choice for work as a trainee or intern.

As for the top management category, it includes positions such as admins, managers, head of departments and executives. The management category also has a decent rating with 4 and 5 being the two most rated. However, it has more lower rated reviews than the trainee category with 1 and 2 being the lower ratings. The position is still highly recommended but individuals that are applying for the management category will need to consider the conditions and factors to identify if it is suitable for them.

Lastly, the technical category has its satisfaction level at slightly moderate. This is because it has the least high rated reviews while having the most moderate rated reviews out of the 3 categories. Similarly to the management category, the technical sector as a position can be a

considerable position. However, individuals will have to do their research on their interested position and carefully consider the factors and conditions.

3.3 Summary



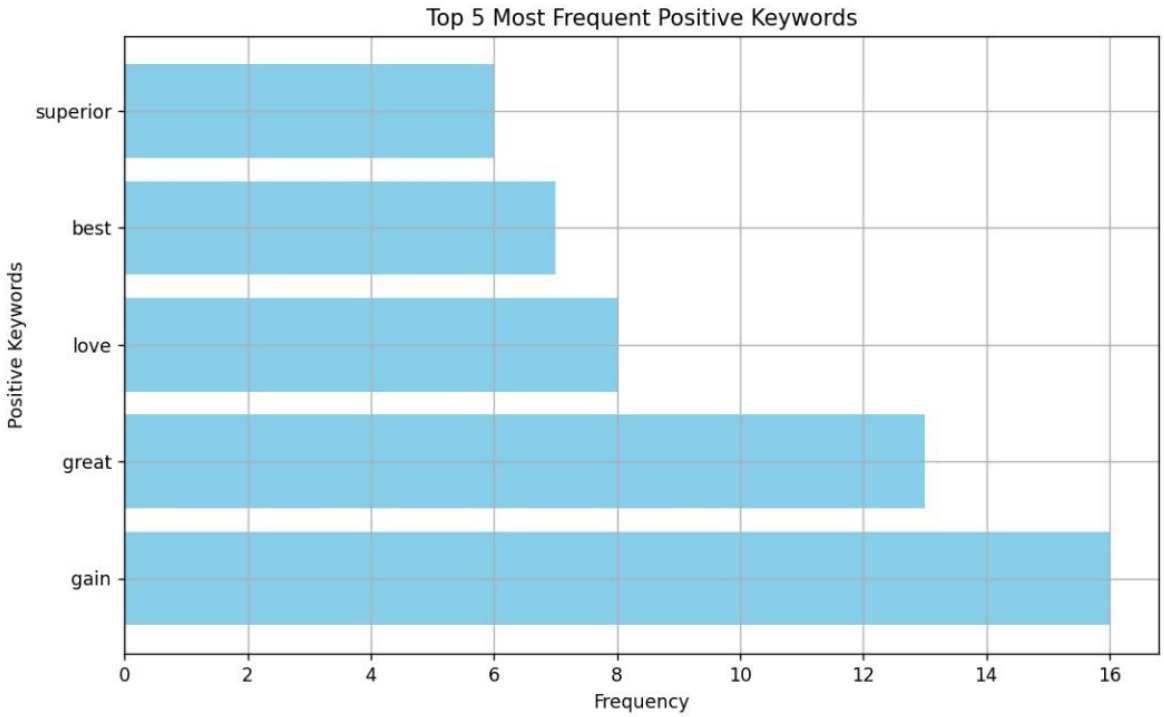
Pie chart is a data visualization to display categorical data by dividing a circle into slices with numerical proportions. Each slice represents a category however, in this finding, slices are used to represent different words. It is used to show the percentage of occurrence of the top 5 words in summary column. It is easy to compare sizes and easy to understand.

Analyzing the summary column is expected to find out the top 5 words that are the most frequent from the 200 scrapped data. This pie chart is generated based on the top 5 words that repeats the most throughout the 200 reviews. The words experience has appeared 26.1% times, good has appeared 23.9%, work has appeared 19.9%, great 15.9% and place 14.2%. The percentage is calculated by dividing the number of times it appeared over the total number of the top 5 repetitive words. This concludes that in the 200 reviews, most people mentioned the word experience.

From the pie chart, the word “experience” appeared the most in the summary column. This may be due to employees describing their overall working experience, their satisfaction or dissatisfaction with their experience, reflection on their experience working with different leaders and many more. If the reviews containing this word are predominantly good, the possible reasons could be positive work environment, they are satisfied with their job, working experience such as the workload given, working with leaders, and management is good. However, if it is predominantly bad, it indicates issues with the company overall environment, culture, working styles and more.

The word “good” scored 23.9% among the five top repetitive words which is a positive indicator as good is a positive word. This can indicate that the feedbacks are suggesting that the employees are satisfied with the company, the company culture is positive, and good management.

3.4 Good



The horizontal bar chart as compared to other modes of data presentation gives us the ability to contrast groups. The bars in them represent each category while their length shows the values of the categories hence making it possible to compare different classes directly. Their ease of reading, compactness and adaptability make them a perfect choice for several applications like business analysis and scholarly work.

Top 5 Positive Keywords from Top Glove evaluations to highlight areas where employees rated for satisfaction in company are chosen to represent in this plot which is “superior”, “best”, “love”, “great”, and “gain”, with their respective frequencies values of 6, 7, 8, 13 and 16. In this scenario, horizontal bar chart naturally lends values in a ranked order to enable a clear ranking which is easy to observe a highest and lowest values at a glance. Y-Axis was labeled as “Top 5 Positive Keywords” with the listed keywords "superior", "best", "love", "great", and "gain" while X-Axis labeled as “Frequency” displays the numerical frequency of each keyword, likely representing how many times each keywords appears. The bars are displayed in descending order of frequency from top to bottom.

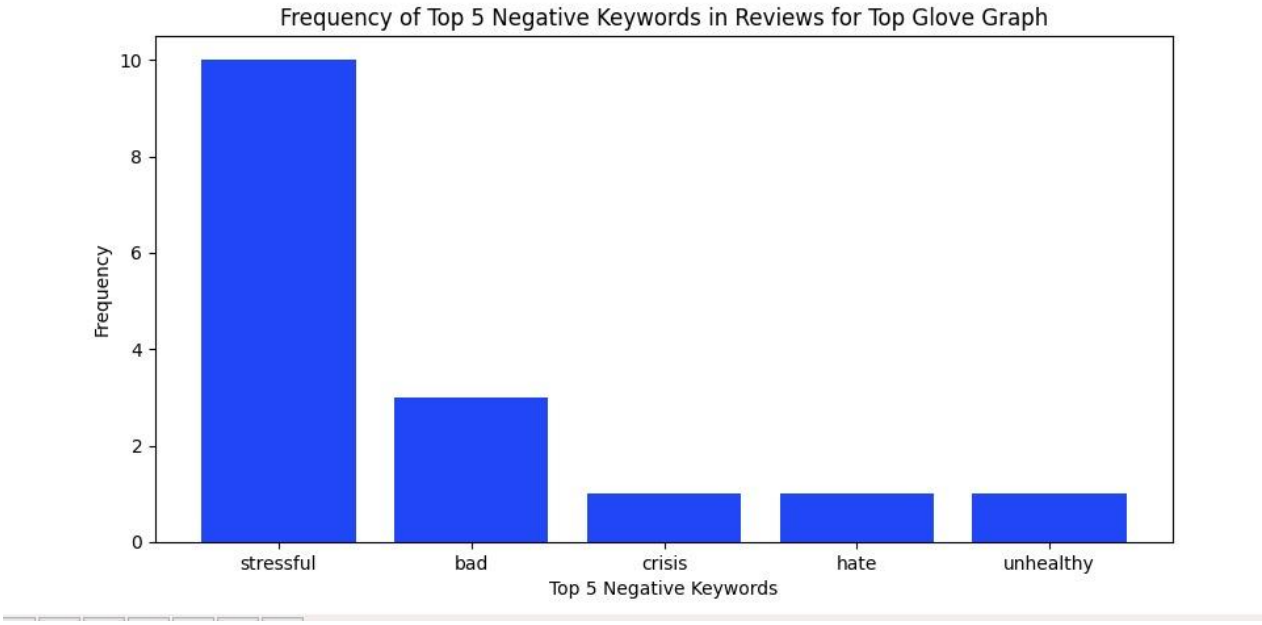
From the insights, we could see that “gain” is the most frequent keyword which occurs approximately 16 times, it shows that employees who worked in this organization gain knowledge, experiences and opportunity exposures associated with improvement and growth; “superior” is the least frequent keyword among the top five with about 6 occurrences. From this we can know that superiors were nice and willing to teach, and it is a concern in a job.

There is a noticeable difference in keyword frequency, with "gain" appearing significantly more frequently than "superior", showing a moderate spread among the top five keywords.

Moreover, we can observe the second most frequent keyword which is “great” with approximately 12 occurrences for the great environment and platform. Top Glove employees enjoyed the company’s benefits with free accessible gym room and flexible working hours. The third most frequent keyword is “love” with approximately 8 occurrences indicating high levels of satisfaction or passion in overall feedback. Last but not least, we have the fourth most frequent keyword which is “best” with approximately seven occurrences implies a higher level of quality and performance. The difference between "gain" (15) and "great" (12) is less than the difference between "great" (12) and "love" (8). The keywords "best" and "superior" have relatively similar frequencies (7 and 6, respectively).

In summary, the plot above effectively visualizes the top five most frequently used positive keywords, providing valuable insights into the dataset's themes and sentiments. The distribution of keyword frequencies indicates a high level of positive feedback, with a focus on growth, quality, and emotional engagement. The chart's design promotes readability and comparison, making it an effective tool for data analysis. The keyword “gain” has the highest frequency among the top 5 positive keywords. Throughout the overall result and data above, Top Glove is viewed positively in this dataset.

3.5 Challenges



A bar chart has been chosen to represent the frequency of the top 5 negative keywords from the Top Glove evaluations to highlight areas where employees are not satisfied with the organization. Our purpose in examining the chart is to provide a comprehensible and graphic depiction of the most prevalent negative sentiments that employees have voiced in their comments. The keywords “stressful”, “bad”, “crisis”, “hate” and “unhealthy”, with their respective frequencies of 10, 3, 1, 1 and 1, denote different elements of the workplace that workers find problematic. One of the reasons for using bar charts is because they can focus on important data points and are aesthetically appealing. In this case, the noticeably higher threshold for “stressful” compared to the others calls immediate attention to the primary concern among employees. Another reason is that its vertical bars effectively display the variations in frequency between keywords.

From this bar chart, we can observe that the term "stressful" stands out significantly, with a much higher frequency than the other keywords. This prominent appearance of "stressful" implies that stress is a predominant issue among Top Glove employees. Employee stress levels this high could be a sign of a few underlying issues, including tight deadlines, overly demanding workloads, a lack of resources or a high-pressure working environment. These factors may result in exhaustion, a decline in job satisfaction, and lower overall staff morale, all of which may have an impact on their productivity and the organization’s bottom line.

Even if they are less frequent, the other keywords nevertheless draw attention to particular and important worries among staff members. The keyword "bad," appearing 3 times, denotes a general feeling of unhappiness or negative encounters with different areas of the job or workplace. This could include a variety of problems, such as unproductive management, a dearth of possibilities for professional growth, poor communication, and a lack of supervisor support. Similarly, the keyword “crisis” albeit less common, points to certain occurrences or periods of extreme challenges or disturbances within the organization. This could be related to problems with finances or organizational changes that have negatively influenced workers, making the working environment turbulent and causing anxiety, uncertainty among employees.

The keyword “hate” while less frequently used, shows a strong negative emotion that stems from a deep-seated dissatisfaction or hatred towards one’s employment or workplace. This intense feeling may have many causes, such as toxic work culture, perceived injustices and unfair treatment from superiors or colleagues. It emphasizes the urgency of addressing and resolving these issues right away. Last but not least, the keyword “unhealthy” which only appears once, raises concerns regarding the mental or physical health effects of the working circumstances at Top Glove. This can relate to dangerous work sites, insufficient health and safety precautions, or psychological troubles brought on by the work atmosphere.

In summary, the bar chart offers a clear visual representation of the key negative sentiments expressed by Top Glove’s employees. The phrase “stressful” is used the most, which implies that the company should prioritize treating stress and its root causes. Meanwhile, other negative sentiments like “bad”, “crisis”, “hate” and “unhealthy” draw attention to further areas of concern that need targeted interventions. Through comprehension and resolution of these concerns, Top Glove can endeavour to establish a more fulfilling and supportive work environment for its staff members.

4.0 Conclusion

4.1 Reflections (Chia Wan Ying)

The task was to scrape reviews from a website, process the text data to extract meaningful insights, and visualize some of those insights using plots. This involved several steps, including web scraping, data processing and plotting. I got to know more about the website JobStreet and how does the review go in Top Glove throughout this project. I have learnt how to create a plot with coding in Python, I found it challenging at first as I spend a lot of time in deciding a suitable plot to use with and some error exists when I run the module as choosing the right type of plot to have a better visualization and ensuring accurately. The final output was visualized successfully with the insights derived from the reviews.

As technology advances, so will the opportunities and challenges that coding brings. Looking forward, I feel like coding is more than a tool, yet it is a process of discovery and development. It enables me to build, create, and innovate in ways that were previously unimaginable. As I continue to navigate in this field, I am grateful for the lessons learned, the challenges overcome, and the limitless opportunities that coding provides.

4.2 Reflections (Tai Yong Xuan)

The purpose of this project is to allow us to learn how to scrape data from websites, filter data, and plot graphs and charts from the scraped data. In this project, I have contributed on coding parts of the program. I have learnt how to use selenium to automate web browser to perform web scraping and natural language toolkit (NLTK) to tokenize text, remove stopwords, and POS tagging. Moreover, I have explored more plots from matplotlib library. Lastly, I have learnt to inspect a website and locate each element through their CSS selector. At the beginning of this project, I have faced difficulties in extracting data from JobStreet. Initially I used BeautifulSoup to scrape data however JobStreet contents are dynamic thus I learned to use Selenium in order to scrape dynamic contents. Not only that, I have also applied the python skills I learnt earlier such as while loop, defining a function, pandas DataFrame, and creating bar chart. In terms of improvement, I am able to identify errors and fix it. Instead of coding directly, I learnt to plan and break down the project into smaller tasks so that it is easier to locate the errors. The project went smoothly, and I have completed my part on time. The program written is able to scrape data, filter it based on the specified criteria and create plots from the scraped data. By completing this project, I have strengthened my programming skills, it also inspired me to learn more and explore more in programming.

4.3 Reflections (Ooi Shi Qi)

When I first started this project, I thought about representing the top negative keywords from the Top Glove employee feedback using a radar chart. The radar chart's capacity to show several variables and their values in a single, well-designed layout made it seem enticing. However, I decided against it since radar charts are better suited for cross-category comparisons than they are for highlighting frequency variations within a single dimension. The overlapping regions and complex visuals could have made it difficult to interpret the distinct variations in keyword frequencies.

The choice of a bar chart turned out to be wise since the vertical bars clearly showed the sharp difference in the frequency of the term "stressful" compared to the other keywords, indicating that stress is the most important problem among workers. This experience made it even more evident how crucial it is to choose the right visualisation technique to successfully and clearly convey data insights.

Working on this assignment was both challenging and rewarding. It took perseverance and patience to handle web scraping with Selenium, particularly while traversing pages and handling dynamic content. Implementing data analysis using the NLTK and pandas libraries in Python further tested my programming skills, because it involved a thorough understanding of text processing and sentiment analysis. Despite the challenges, working on this project has really enhanced my confidence in my ability to evaluate and conceive data using Python. Moving forward, I recognize the value of critically assessing different chart types and their suitability for specific datasets and analysis goals, and I am eager to explore more advanced techniques to derive deeper insights.

4.4 Reflections (Terrence Teoh Jin Haw)

Over the course of completing this project, our group has encountered multiple issues when it comes to web scraping the data. It took multiple attempts to overcome this issue, but with the

help of the feedback and advice from the lecturer, and the hard work of our group, we were able to overcome the issues and complete the project. While we were progressing on our project, I have acknowledged how important communication is. It made decision making easier and, questions and doubts solved without any concern. Other than that, opinions can be easily brought and discussed. With that, the result of the project can be at its best form when ideas from everyone are thrown and complied together. By acknowledging this, communication will be prioritized while moving forward to other projects.

There are multiple things I have learned throughout the progress of this project. We were given an opportunity explore more plots with using python. It was a learning experience learning how much plots that can be coded through python and learning on how to code the plots.

I was tasked with tackling the position variable. Initially, I was concerned on what type of plot would go well with the position data as the position data is not generated by the company, instead it was written by the reviewers. The reviewers could have written random position of work and data will need to be filtered. With that, I thought of an idea, I filtered the data by using keywords to categorize the position of work and have the rating variable being my secondary variable. I chose the swarm plot as the choice for my analysis and figured I would analyse the satisfaction level of each category of position. It was a tough progress but it was truly a learning experience.

4.5 Conclude findings

In conclusion, we have gathered deep insights into the feelings and the level of satisfaction of employees working at Top Glove. Overall, Top Glove is towards being a positively reviewed and perceived company, being an opportunity for individuals as it provides various benefits

and gains for the employees of Top Glove. As working experience and working environment being the two most crucial factors, individuals should make careful consideration when applying for a position at Top Glove. If individuals are interested, it is highly recommended for those interested in working as a trainee or in the top management sector of Top Glove. However, from our findings, it can be stressful to work at Top Glove due to the huge amount of workload and the high pressure of the working environment. Therefore, individuals will need to carefully evaluate their decision when applying for a job at Top Glove.

Apart from that, Top Glove's management should also prioritize making targeted improvements, so that the overall employee performance and satisfaction can be boosted. It is imperative to address the mixed experiences of workers in technical and managerial roles, which calls for a deep understanding of the specific challenges and areas of concern that fall into each of these categories. Workplace stressors including strict deadlines, heavy workloads, and high-pressure settings should be identified and mitigated in order to lower stress levels. Furthermore, it is essential that management persist in cultivating and broadening growth prospects, given that the regular reference to affirmative terms such as "gain" suggests that staff members place a premium on the skills and experiences they pick up at Top Glove. By building on the positive aspects of the work culture and resolving specific negative issues like organizational crises, health-related concerns and negative emotions, Top Glove can create a more supportive and fulfilling work environment that attracts and retains top talent.

4.6 Elements(s) not able to complete

One element we were not able to figure out and complete before turning in the assignment was the scraping JobStreet's job vacancy posts. The scraping process became challenging as the job vacancy elements did not have specific class names, IDs or selectors that could have been easier for scraping. The difficulty arose when we were trying to isolate and extract only the relevant data from the web pages due to the lack of consistent and unique identifiers. Additionally, we encountered issues with extracting salary information. This is due to the inconsistently formatted or embedded within other elements, thus complicating the scraping procedure. Consequently, we were unable to obtain an extensive dataset of job vacancy postings and wage details, which limits our capacity to analyze and present these aspects in our report.

4.7 Program bugs

During the coding process of this project, we encountered several program bugs that were difficult to fix and needed careful debugging and troubleshooting.

Def filtering() unable to filter 'i'

One of the problems found was with the filtering() function, which was designed to filter out the word 'i'. Despite our efforts, the function was unable to exclude the lowercase 'i' from the dataset. There are several explanations for why this could have happened. First, the way the tokenization process handles single-character words may be the root of the problem. In the filtering stage, the tokenizer may inadvertently skip the letter 'i' or treats it as a special character. Second, it could be due to the specifics of the filtering function's implementation, where the condition to exclude 'i' might not be correctly integrated. In order to specifically exclude it, we added the condition, word!= 'i', to address this. However, this issue highlighted the difficulty of text preprocessing and the requirement for meticulous attention to detail to guarantee that all superfluous words are successfully filtered out.

Scraped data vary

Another significant challenge was the variability in the data that was scraped. In particular, the mean of the ratings varied occasionally each time the data was scraped. We typically obtained a mean rating of 3.82, but at times, the outcome was different. This variability might be due to the dynamic nature of the content on the JobStreet website, which loads different reviews each time a scrape is executed. The specific set of 200 reviews retrieved depends on what data JobStreet loads for us at that moment.

These differences could be caused by several variables, such as modifications to the website's algorithms for loading reviews, the time of day, the recentness of the reviews, and possible server-side caching techniques. These elements play a part in the dataset's inconsistency, which influences the analytical outcomes. This variation made it clear how crucial it is to comprehend the restrictions and limitations of online scraping, especially when working with content that is generated dynamically.

Reflecting on these bugs, it is evident that while the process of web scraping and data analysis is powerful, it also requires a thorough understanding of the source data and the potential issues that can arise. Despite the difficulties, fixing these errors was a great way to gain experience and reinforced the importance of robust error handling and data validation procedures in programming.

5.0 References

Krukowski, I. (2024, May 13). Web Scraping Tutorial Using Selenium & Python (+ examples). *ScrapingBee*. https://www.scrapingbee.com/blog/selenium-python/

What makes a good star rating for products in different industries. (n.d.). https://www.meetyogi.com/post/what-makes-a-good-star-rating-for-products-in-different-industries#:~:text=A%20rating%20of%203.5%20stars,for%20the%20majority%20of%20customers.