DSCI 560: Data Science Professional Practicum

Laboratory Assignment 4 Part 2

Team Details:

Team Name: Team PVC

Team Member	USC ID
Haoran Zhang	3502885083
Venkatesh Dharmaraj	6773159759
Yichen An	2780765696

Message Content Abstraction

Doc2Vec: This takes in the CSV file from lab4 part 1 and transforms the keyword extracted for that document into vector forms

```python
import pandas as pd
from gensim.models.doc2vec import Doc2Vec, TaggedDocument
from nltk.tokenize import word_tokenize

# Load the CSV file
df = pd.read_csv('posts.csv')

# Assuming 'formatted_post_title' is the column with text data
data = df['formatted_post_title'].tolist()

# Preprocess the documents and create TaggedDocuments, using post_id_tech as tags if needed
tagged_data = [TaggedDocument(words=word_tokenize(doc.lower()), tags=[str(i)]) for i, doc in enumerate(data)]

# Train the Doc2Vec model (adjust parameters as needed)
model = Doc2Vec(vector_size=20, min_count=2, epochs=50)
model.build_vocab(tagged_data)
model.train(tagged_data, total_examples=model.corpus_count, epochs=model.epochs)

# Infer vectors for the original dataset (or any new data)
document_vectors = [model.infer_vector(word_tokenize(doc.lower())) for doc in data]

# Print the document vectors alongside their titles
for i, vec in enumerate(document_vectors):
    print(f"Document: {data[i]}")
    print(f"Vector: {vec}\n")
```

```
Document: In a First, a Prosthetic Limb Can Sense Temperature Like a Living Hand
Vector: [ 0.3332558  -0.17202151 -0.02791681  0.07258766  0.26919416 -0.37276015
 -0.11447767  0.44625977  0.06345354  0.13722219  0.06270309  0.01030525
  0.10246638  0.05064563  0.08841644  0.18558119  0.59672666 -0.03281841
 -0.13928427 -0.02914918]

Document: Alternate qubit design does error correction in hardware. Early-stage technology has the potential to cut qubits needed for useful computers.
Vector: [-0.3256915   0.21789767 -0.18155953 -0.03502605  0.40092766 -0.40997815
  0.23175985  0.43988124  0.01394675 -0.11393522  0.130297   -0.06024748
 -0.11098453  0.33694077  0.29514936  0.55408835  0.5093762   0.06292582
 -0.43755668 -0.64656985]

Document: Gel and lithium-ion tech could enable 1000-mile EV range on one charge | Researchers achieve EV battery breakthrough with silicon-based materials and gel electrolytes, moving closer to a 1,000-ki
lometer range on a single charge.
Vector: [ 0.04483001  0.05117815  0.08749112  0.2614808  -0.10640705 -0.48808905
 -0.11357064  0.6323668   0.22529991  1.2392393  -0.23424962  0.5005762
 -0.24022935 -0.37916362  1.0229969   0.80105174  0.52243483 -0.7136255
  0.2907413  -0.0608465 ]

Document: Inhalable sensors could enable early lung cancer detection
Vector: [-0.13791318  0.0185714  -0.49091372  0.24483877  0.18482071  0.1372347
  0.3038938   0.14694436 -0.21829017  0.55681986  0.16870919 -0.22888522
  0.11537617 -0.2140774   0.5503922  -0.03265684  0.2816531  -0.48537165
 -0.11538043  0.0152991 ]

Document: Scientists develop a low-cost device to make cell therapy safer | A plastic microfluidic chip can remove some risky cells that could potentially become tumors before they are implanted in a patie
nt.
Vector: [ 0.6079858  -0.13758685 -0.517426   -0.07735139 -0.6894494  -0.10371833
 -0.35433185  0.3665978  -0.45838818  1.0604633  -0.35693288  0.1846211
  0.39967406 -0.04315176  1.1437122   0.26860696  1.0905862  -1.1873078
 -0.05681656  0.22932011]
```

Clustering Messages

The inferred vectors are then used as input to a clustering algorithm, such as K-means. K-means clustering aims to partition the set of vectors into K clusters, with each document assigned to the cluster with the nearest mean vector. This clustering step groups together documents with similar meanings based on their vector representations.

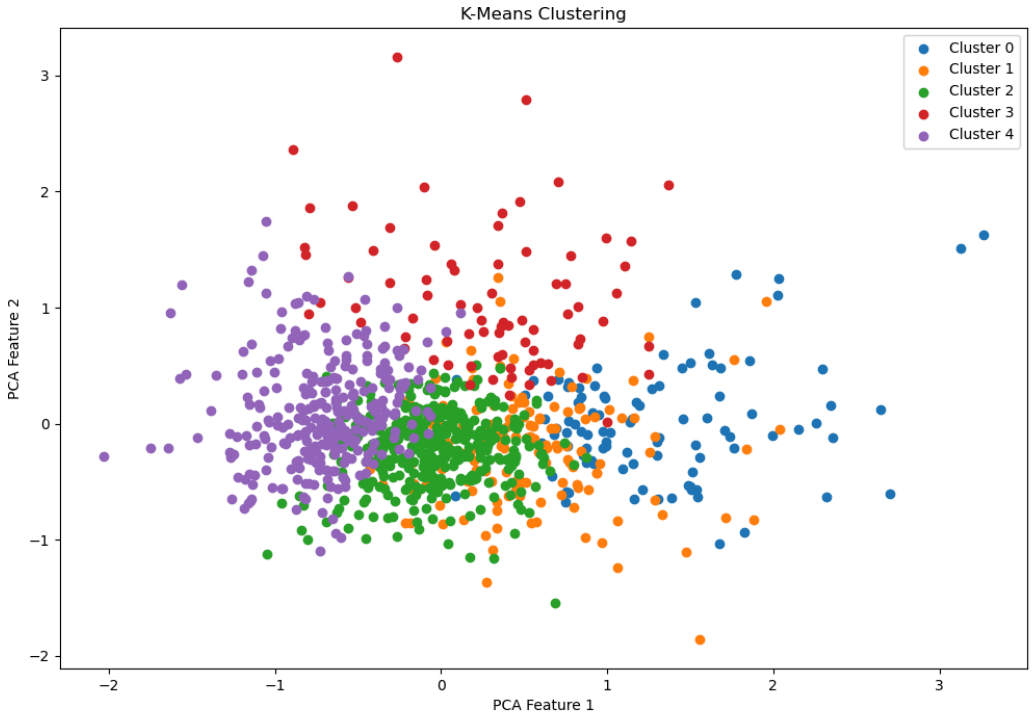
Clustering.py

```python
# Print Cluster
print(clustered_data)

import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans

# Clustering
kmeans = KMeans(n_clusters=5, random_state=42)
kmeans.fit(X)
labels = kmeans.labels_

# Deploy PCA
pca = PCA(n_components=2)
X_r = pca.fit_transform(X)

# Create matplotlib graphs
plt.figure(figsize=(12, 8))

for i in range(5):
    plt.scatter(X_r[labels == i, 0], X_r[labels == i, 1], label=f'Cluster {i}')

plt.legend()

plt.title('K-Means Clustering')
plt.xlabel('PCA Feature 1')
plt.ylabel('PCA Feature 2')
plt.show()
```

After clustering, the results can be analyzed to understand the common themes or topics within each cluster. Each cluster should ideally represent a collection of documents that are semantically similar to each other.

	document	cluster
0	This ultrasound sticker senses changing stiffn...	0
1	In a First, a Prosthetic Limb Can Sense Temper...	2
2	Alternate qubit design does error correction i...	4
3	Scientists develop a low-cost device to make c...	0
4	Gel and lithium-ion tech could enable 1000-mil...	0
..	...	...
992	The clock speed wars are back as Intel brags a...	2
993	Nuclear fusion reactor in Korea reaches 100 mi...	4
994	Twitter whistleblower cites security flaws bef...	2
995	Two atomic clocks have been quantum entangled ...	3
996	Nuclear fusion reactor in Korea reaches 100 mi...	4

For better interpretability, the high-dimensional vectors can be visualized in two or three dimensions using techniques like t-SNE (t-distributed Stochastic Neighbor Embedding). This helps in understanding the distribution and separation of the different clusters.



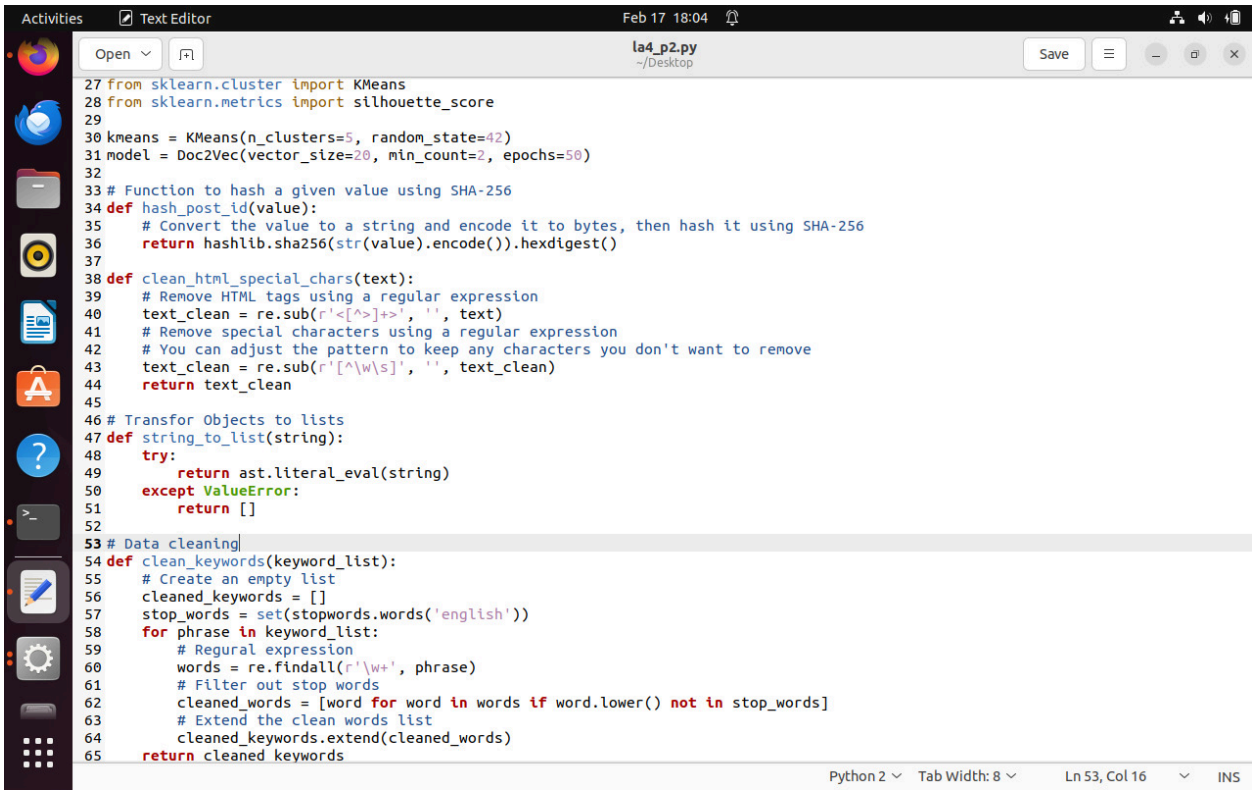
Evaluation of the Algorithm: The K-means clustering algorithm is been evaluated based on the ***silhouette score***.

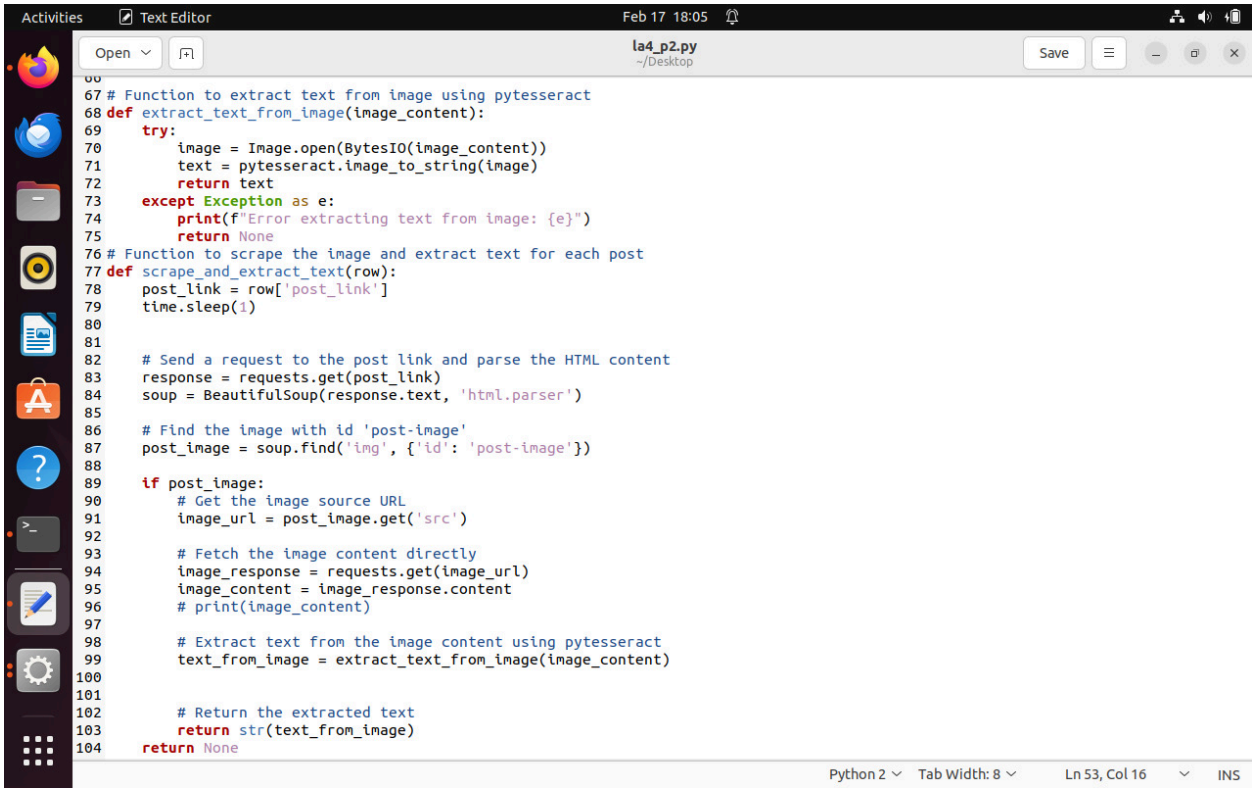
Automation

- The scripts of Part I of the assignment and the clustering algorithm are combined as a single compiled script. The logic for automation to perform the mentioned tasks is also added to the script.
- To execute the script, run the following command:
    - python3 la4_p2.py <interval_minutes>
    - <interval_minutes> can be any integer
- The script will fetch, preprocess, cluster, and store the data in the database periodically based on the ‘interval_minutes’ value the user gives.
- *Storing Data in the Database*: The script will check whether the post already exists in the database using the post_id value. If exists, It updates the existing record. Else, it creates a new record in the database table.

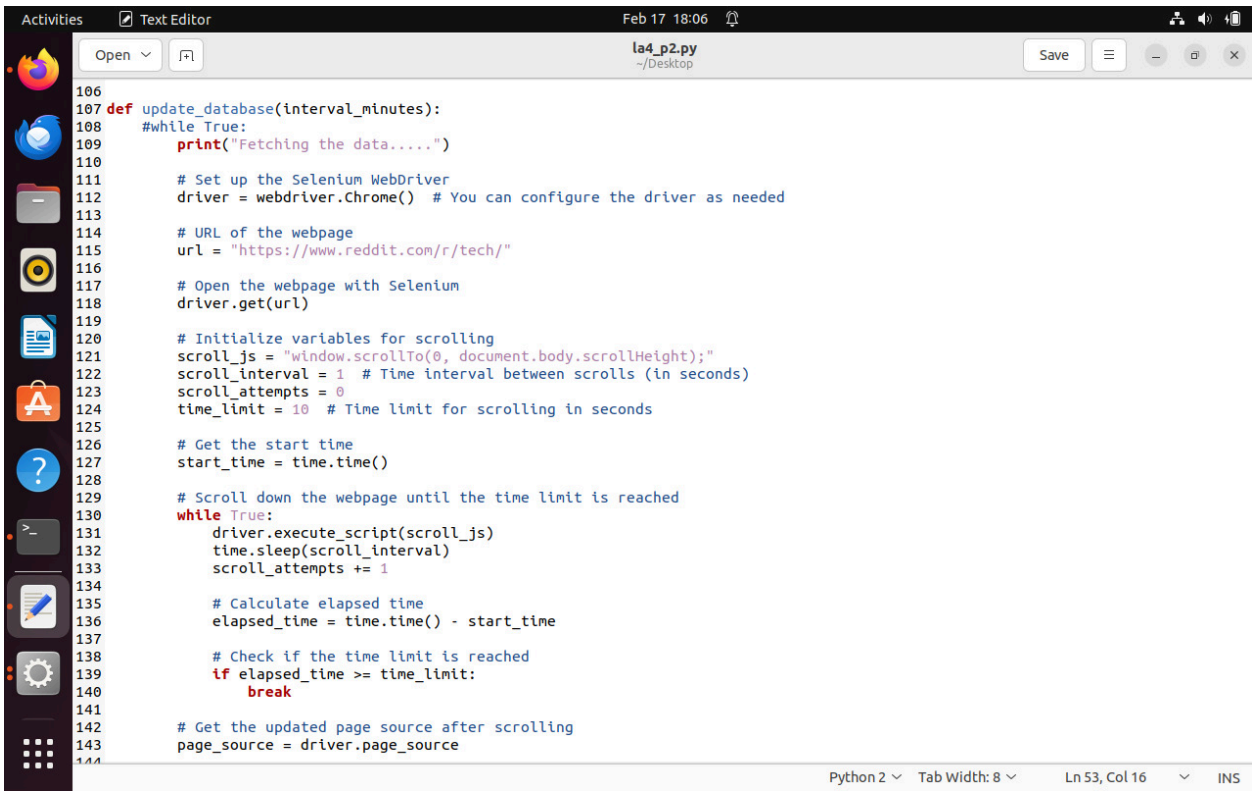
- When the script is not updating the database in the background, the command line prompt will take keywords or a message as input and find the cluster that matches the input closest to it. The messages from a selected cluster should be displayed along with a graphical representation.
- The user needs to input ‘quit’ to stop the execution.
- In addition to the terminal outputs, the script also generates 3 CSVs:
    - posts.csv - contains scrapped post records
    - processed_data.csv - contains processed post records
    - clustered_data.csv - contains post and assigned cluster details

The script is shown in the below screenshots,

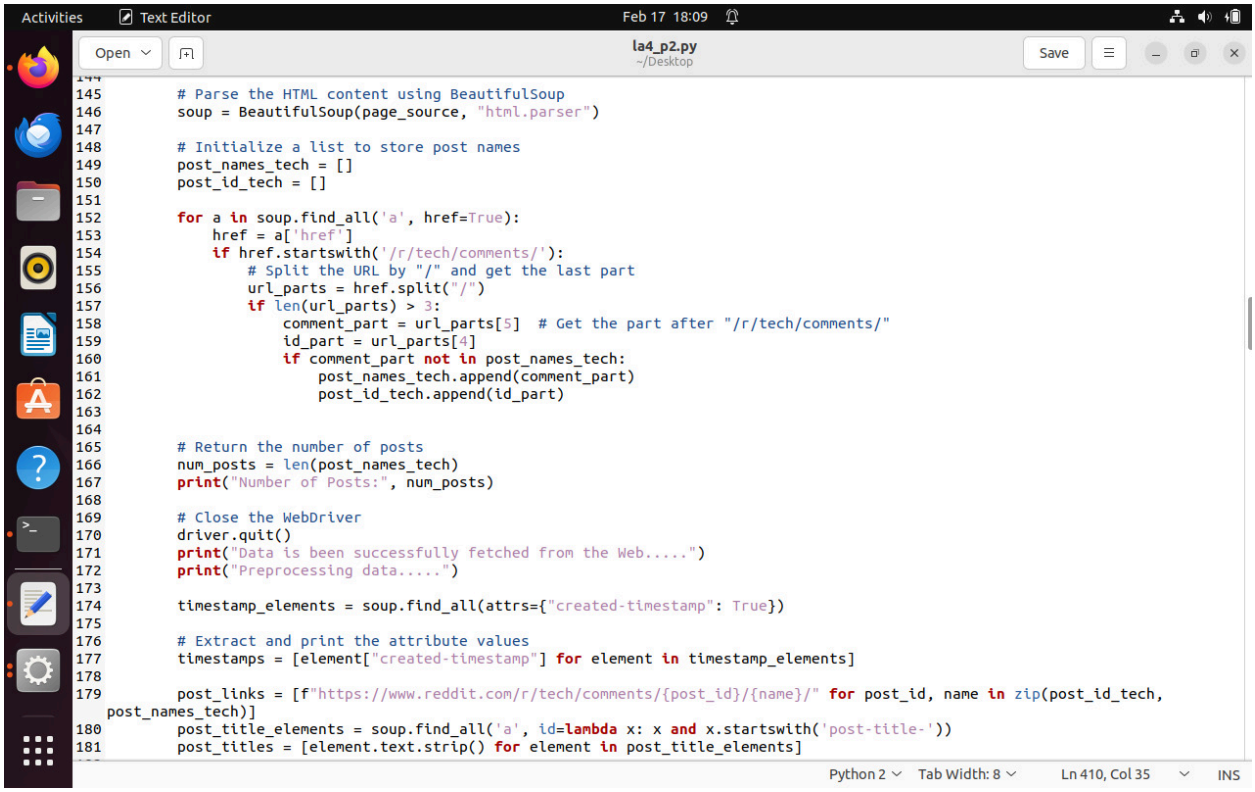


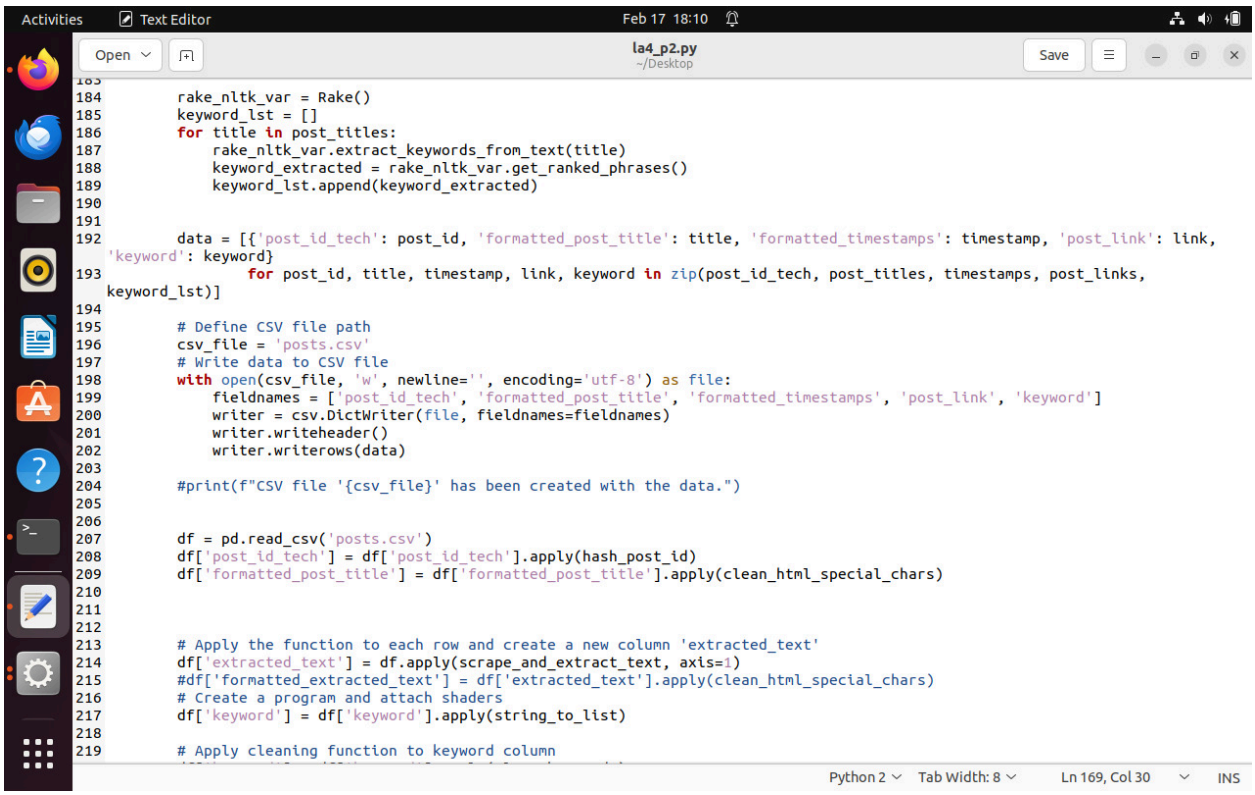


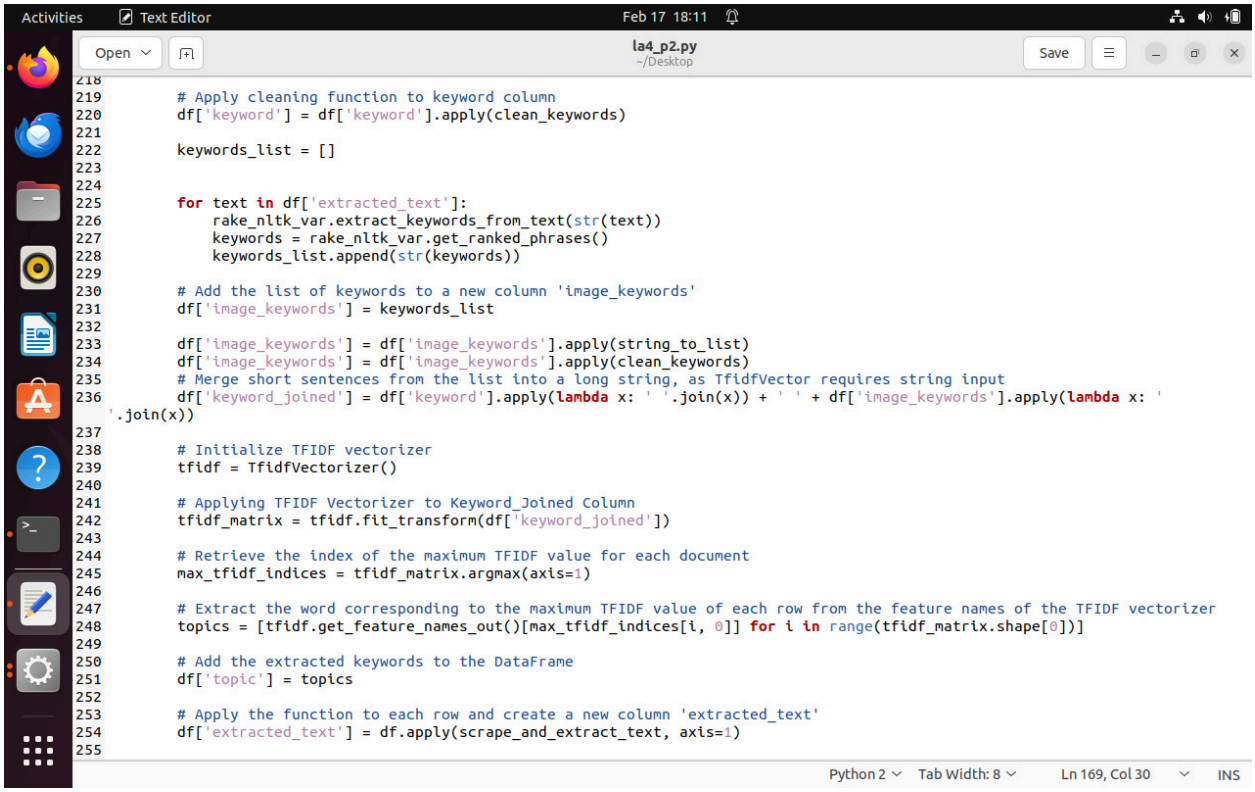

```python
# Function to extract text from image using pytesseract
def extract_text_from_image(image_content):
    try:
        image = Image.open(BytesIO(image_content))
        text = pytesseract.image_to_string(image)
        return text
    except Exception as e:
        print(f"Error extracting text from image: {e}")
        return None
# Function to scrape the image and extract text for each post
def scrape_and_extract_text(row):
    post_link = row['post_link']
    time.sleep(1)


    # Send a request to the post link and parse the HTML content
    response = requests.get(post_link)
    soup = BeautifulSoup(response.text, 'html.parser')

    # Find the image with id 'post-image'
    post_image = soup.find('img', {'id': 'post-image'})

    if post_image:
        # Get the image source URL
        image_url = post_image.get('src')

        # Fetch the image content directly
        image_response = requests.get(image_url)
        image_content = image_response.content
        # print(image_content)

        # Extract text from the image content using pytesseract
        text_from_image = extract_text_from_image(image_content)


        # Return the extracted text
        return str(text_from_image)
    return None
```

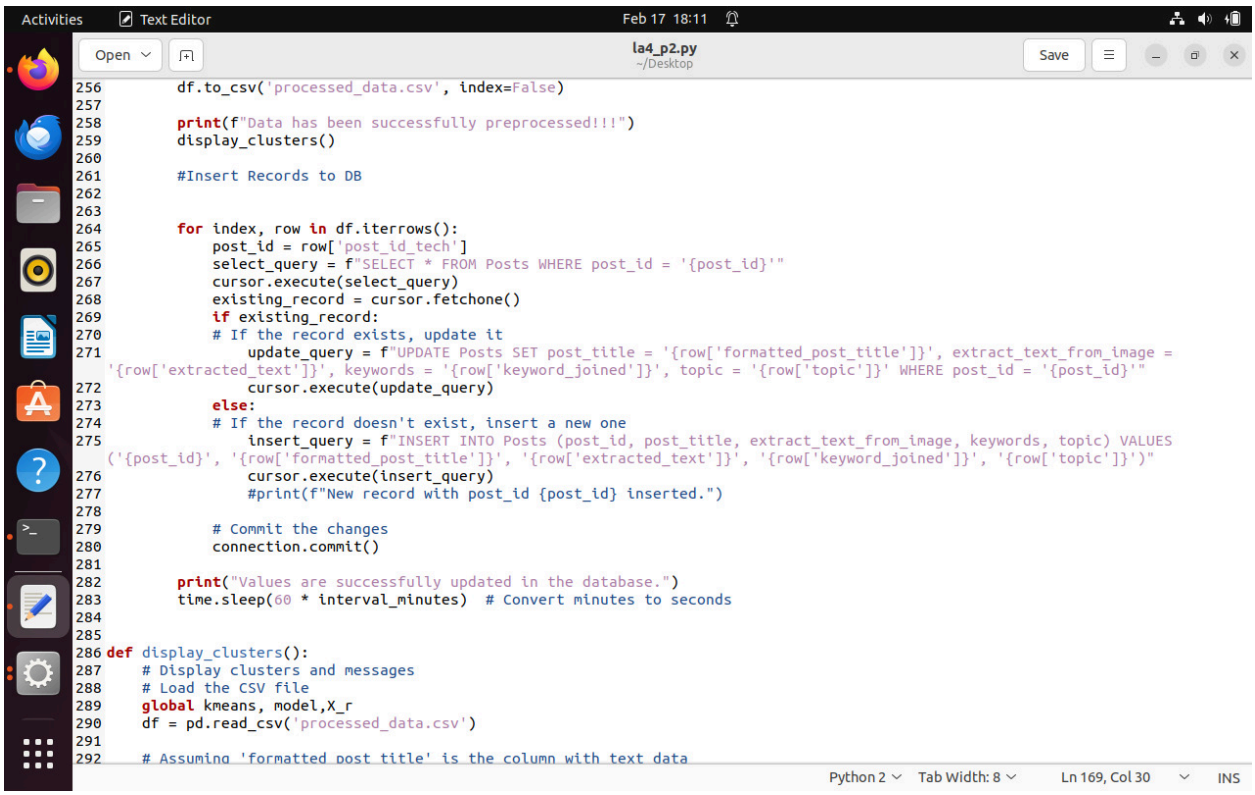

```python

def update_database(interval_minutes):
    #while True:
        print("Fetching the data.....")

        # Set up the Selenium WebDriver
        driver = webdriver.Chrome()  # You can configure the driver as needed

        # URL of the webpage
        url = "https://www.reddit.com/r/tech/"

        # Open the webpage with Selenium
        driver.get(url)

        # Initialize variables for scrolling
        scroll_js = "window.scrollTo(0, document.body.scrollHeight);"
        scroll_interval = 1  # Time interval between scrolls (in seconds)
        scroll_attempts = 0
        time_limit = 10  # Time limit for scrolling in seconds

        # Get the start time
        start_time = time.time()

        # Scroll down the webpage until the time limit is reached
        while True:
            driver.execute_script(scroll_js)
            time.sleep(scroll_interval)
            scroll_attempts += 1

            # Calculate elapsed time
            elapsed_time = time.time() - start_time

            # Check if the time limit is reached
            if elapsed_time >= time_limit:
                break

        # Get the updated page source after scrolling
        page_source = driver.page_source
```


Activities
Text Editor
Feb 17 18:09
Open
la4_p2.py
~/Desktop
Save
# Parse the HTML content using BeautifulSoup
soup = BeautifulSoup(page_source, "html.parser")
# Initialize a list to store post names
post_names_tech = []
post_id_tech = []
for a in soup.find_all('a', href=True):
href = a['href']
if href.startswith('/r/tech/comments/'):
# Split the URL by "/" and get the last part
url_parts = href.split("/")
if len(url_parts) > 3:
comment_part = url_parts[5] # Get the part after "/r/tech/comments/"
id_part = url_parts[4]
if comment_part not in post_names_tech:
post_names_tech.append(comment_part)
post_id_tech.append(id_part)
# Return the number of posts
num_posts = len(post_names_tech)
print("Number of Posts:", num_posts)
# Close the WebDriver
driver.quit()
print("Data is been successfully fetched from the Web.....")
print("Preprocessing data.....")
timestamp_elements = soup.find_all(attrs={"created-timestamp": True})
# Extract and print the attribute values
timestamps = [element["created-timestamp"] for element in timestamp_elements]
post_links = [f"https://www.reddit.com/r/tech/comments/{post_id}/{name}/" for post_id, name in zip(post_id_tech, post_names_tech)]
post_title_elements = soup.find_all('a', id=lambda x: x and x.startswith('post-title-'))
post_titles = [element.text.strip() for element in post_title_elements]
Python 2
Tab Width: 8
Ln 410, Col 35
INS

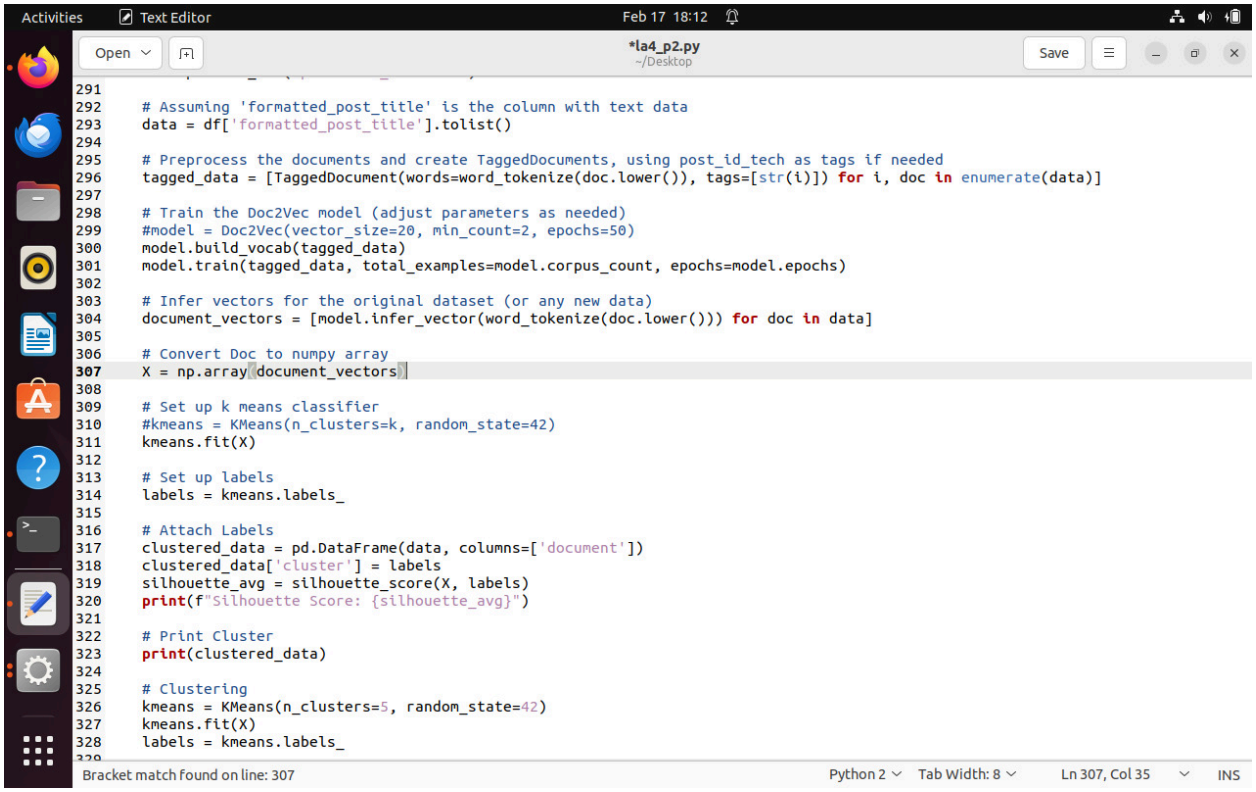

Activities
Text Editor
Feb 17 18:10
Open
la4_p2.py
~/Desktop
Save
rake_nltk_var = Rake()
keyword_lst = []
for title in post_titles:
rake_nltk_var.extract_keywords_from_text(title)
keyword_extracted = rake_nltk_var.get_ranked_phrases()
keyword_lst.append(keyword_extracted)
data = [{'post_id_tech': post_id, 'formatted_post_title': title, 'formatted_timestamps': timestamp, 'post_link': link, 'keyword': keyword}
for post_id, title, timestamp, link, keyword in zip(post_id_tech, post_titles, timestamps, post_links, keyword_lst)]
# Define CSV file path
csv_file = 'posts.csv'
# Write data to CSV file
with open(csv_file, 'w', newline='', encoding='utf-8') as file:
fieldnames = ['post_id_tech', 'formatted_post_title', 'formatted_timestamps', 'post_link', 'keyword']
writer = csv.DictWriter(file, fieldnames=fieldnames)
writer.writeheader()
writer.writerows(data)
#print(f"CSV file '{csv_file}' has been created with the data.")
df = pd.read_csv('posts.csv')
df['post_id_tech'] = df['post_id_tech'].apply(hash_post_id)
df['formatted_post_title'] = df['formatted_post_title'].apply(clean_html_special_chars)
Apply the function to each row and create a new column 'extracted_text'
df['extracted_text'] = df.apply(scrape_and_extract_text, axis=1)
#df['formatted_extracted_text'] = df['extracted_text'].apply(clean_html_special_chars)
Create a program and attach shaders
df['keyword'] = df['keyword'].apply(string_to_list)
Apply cleaning function to keyword column
Python 2
Tab Width: 8
Ln 169, Col 30
INS

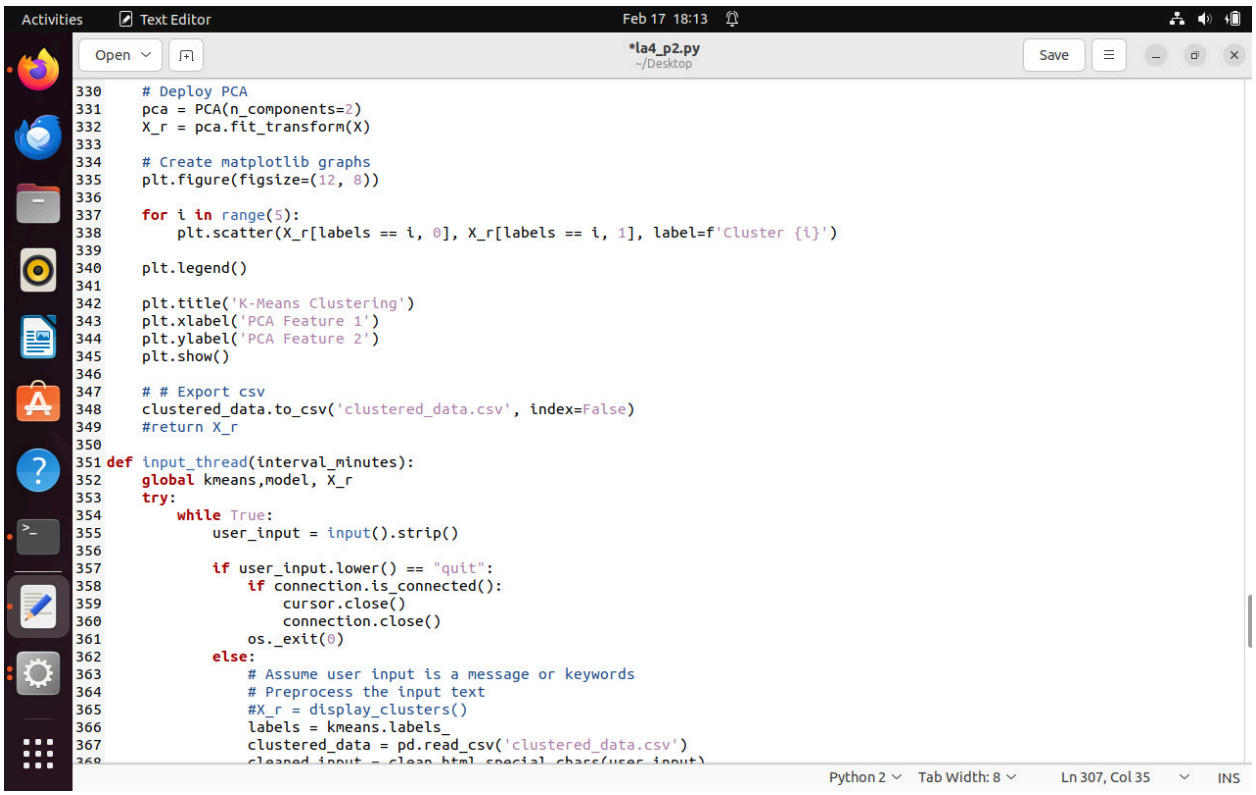


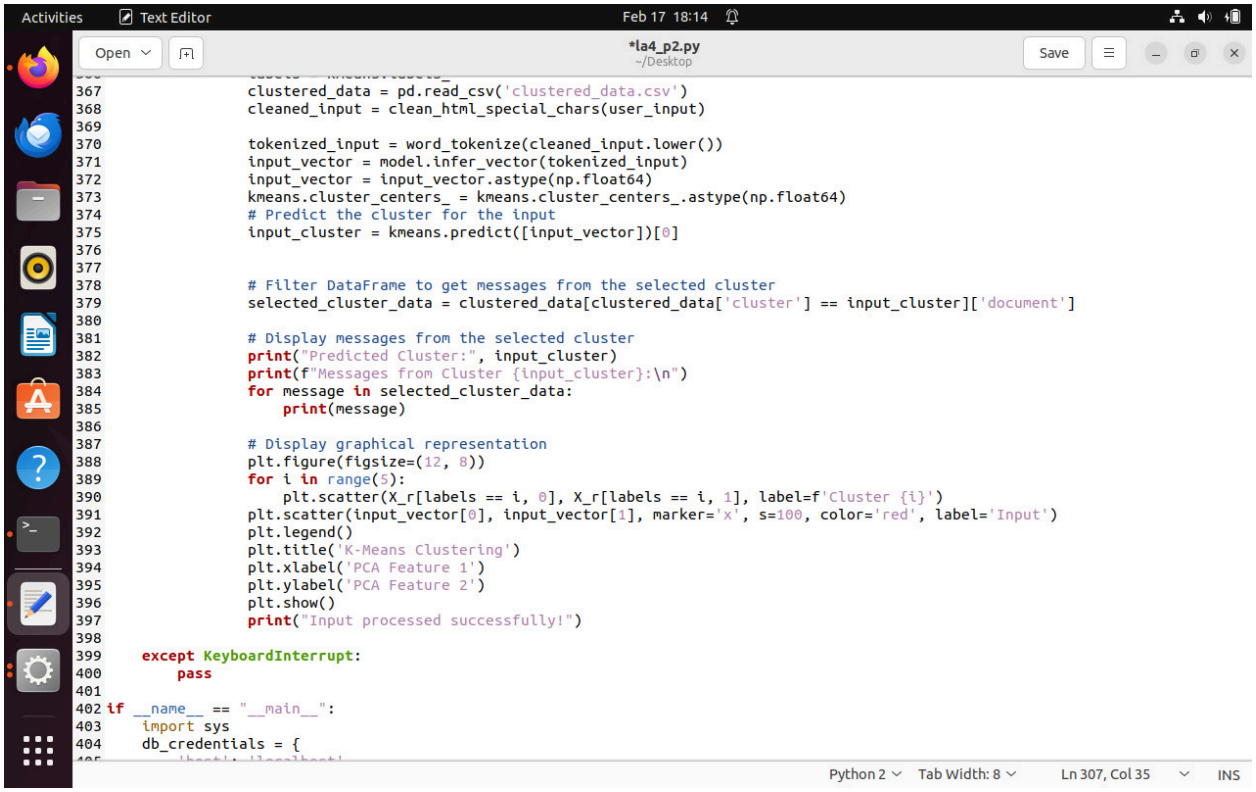

```
218
219         # Apply cleaning function to keyword column
220         df['keyword'] = df['keyword'].apply(clean_keywords)
221
222         keywords_list = []
223
224
225         for text in df['extracted_text']:
226             rake_nltk_var.extract_keywords_from_text(str(text))
227             keywords = rake_nltk_var.get_ranked_phrases()
228             keywords_list.append(str(keywords))
229
230         # Add the list of keywords to a new column 'image_keywords'
231         df['image_keywords'] = keywords_list
232
233         df['image_keywords'] = df['image_keywords'].apply(string_to_list)
234         df['image_keywords'] = df['image_keywords'].apply(clean_keywords)
235         # Merge short sentences from the list into a long string, as TfidfVector requires string input
236         df['keyword_joined'] = df['keyword'].apply(lambda x: ' '.join(x)) + ' ' + df['image_keywords'].apply(lambda x: '
    '.join(x))
237
238         # Initialize TFIDF vectorizer
239         tfidf = TfidfVectorizer()
240
241         # Applying TFIDF Vectorizer to Keyword_Joined Column
242         tfidf_matrix = tfidf.fit_transform(df['keyword_joined'])
243
244         # Retrieve the index of the maximum TFIDF value for each document
245         max_tfidf_indices = tfidf_matrix.argmax(axis=1)
246
247         # Extract the word corresponding to the maximum TFIDF value of each row from the feature names of the TFIDF vectorizer
248         topics = [tfidf.get_feature_names_out()[max_tfidf_indices[i, 0]] for i in range(tfidf_matrix.shape[0])]
249
250         # Add the extracted keywords to the DataFrame
251         df['topic'] = topics
252
253         # Apply the function to each row and create a new column 'extracted_text'
254         df['extracted_text'] = df.apply(scrape_and_extract_text, axis=1)
255
```

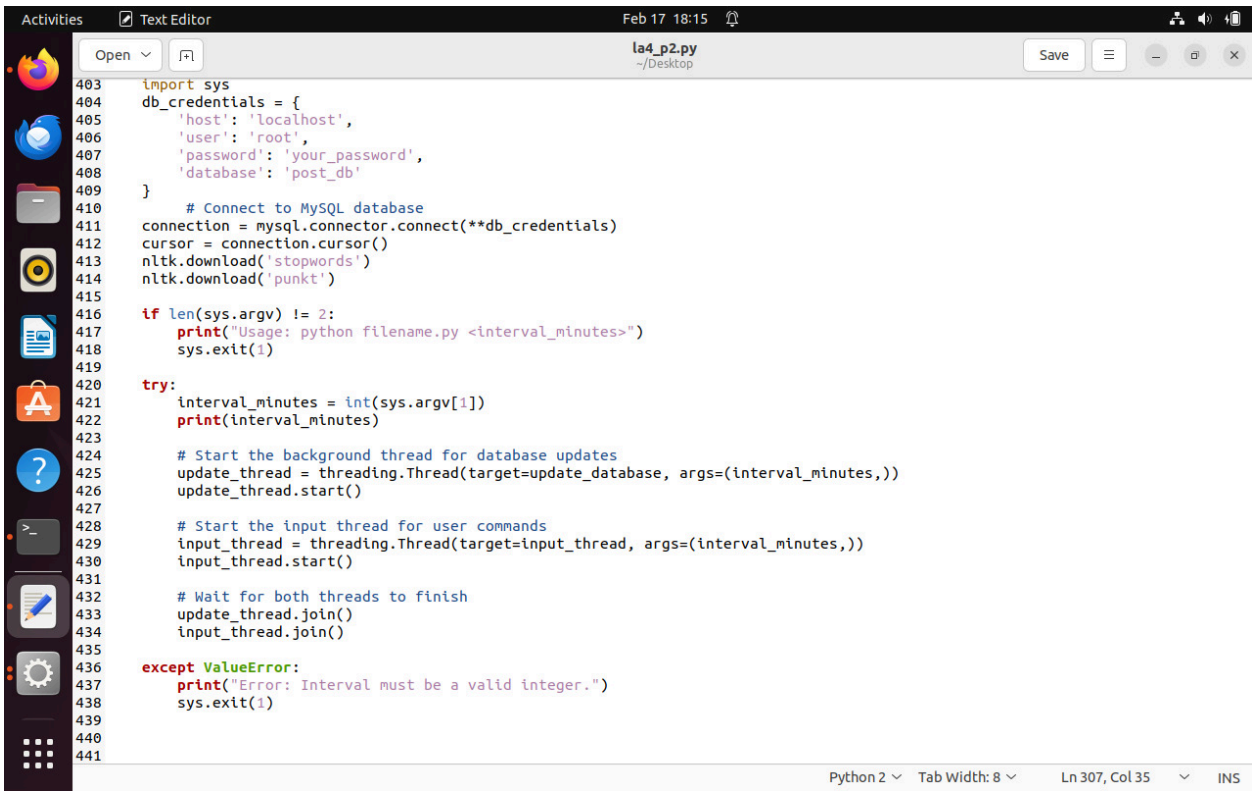




```
256         df.to_csv('processed_data.csv', index=False)
257
258         print(f"Data has been successfully preprocessed!!!")
259         display_clusters()
260
261         #Insert Records to DB
262
263
264         for index, row in df.iterrows():
265             post_id = row['post_id_tech']
266             select_query = f"SELECT * FROM Posts WHERE post_id = '{post_id}'"
267             cursor.execute(select_query)
268             existing_record = cursor.fetchone()
269             if existing_record:
270             # If the record exists, update it
271                 update_query = f"UPDATE Posts SET post_title = '{row['formatted_post_title']}', extract_text_from_image =
    '{row['extracted_text']}', keywords = '{row['keyword_joined']}', topic = '{row['topic']}' WHERE post_id = '{post_id}'"
272                 cursor.execute(update_query)
273             else:
274             # If the record doesn't exist, insert a new one
275                 insert_query = f"INSERT INTO Posts (post_id, post_title, extract_text_from_image, keywords, topic) VALUES
    ('{post_id}', '{row['formatted_post_title']}', '{row['extracted_text']}', '{row['keyword_joined']}', '{row['topic']}')"
276                 cursor.execute(insert_query)
277                 #print(f"New record with post_id {post_id} inserted.")
278
279             # Commit the changes
280             connection.commit()
281
282         print("Values are successfully updated in the database.")
283         time.sleep(60 * interval_minutes)  # Convert minutes to seconds
284
285
286 def display_clusters():
287     # Display clusters and messages
288     # Load the CSV file
289     global kmeans, model,X_r
290     df = pd.read_csv('processed_data.csv')
291
292     # Assuming 'formatted_post_title' is the column with text data
```

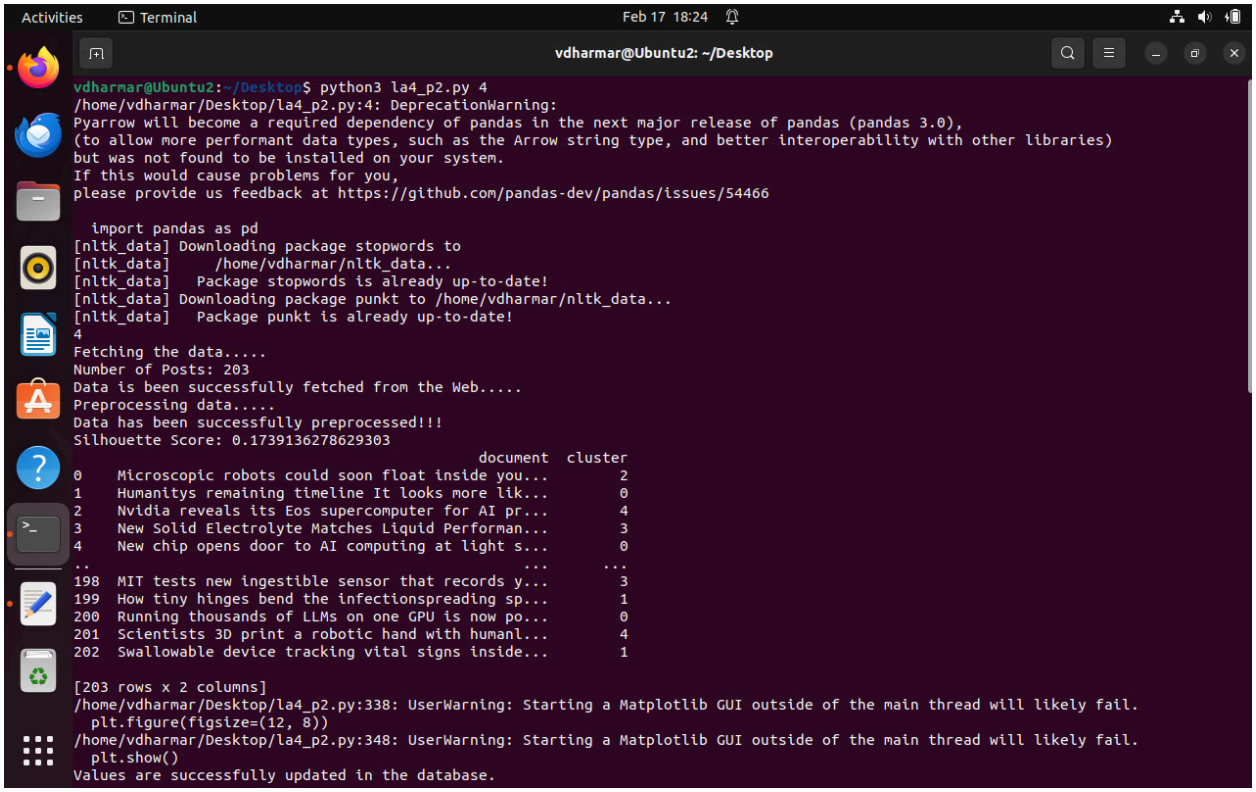

Activities
Text Editor
Feb 17 18:12
Open
*la4_p2.py
~/Desktop
Save
# Assuming 'formatted_post_title' is the column with text data
data = df['formatted_post_title'].tolist()
# Preprocess the documents and create TaggedDocuments, using post_id_tech as tags if needed
tagged_data = [TaggedDocument(words=word_tokenize(doc.lower()), tags=[str(i)]) for i, doc in enumerate(data)]
# Train the Doc2Vec model (adjust parameters as needed)
#model = Doc2Vec(vector_size=20, min_count=2, epochs=50)
model.build_vocab(tagged_data)
model.train(tagged_data, total_examples=model.corpus_count, epochs=model.epochs)
# Infer vectors for the original dataset (or any new data)
document_vectors = [model.infer_vector(word_tokenize(doc.lower())) for doc in data]
# Convert Doc to numpy array
X = np.array(document_vectors)
# Set up k means classifier
#kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(X)
# Set up labels
labels = kmeans.labels_
# Attach Labels
clustered_data = pd.DataFrame(data, columns=['document'])
clustered_data['cluster'] = labels
silhouette_avg = silhouette_score(X, labels)
print(f"Silhouette Score: {silhouette_avg}")
# Print Cluster
print(clustered_data)
# Clustering
kmeans = KMeans(n_clusters=5, random_state=42)
kmeans.fit(X)
labels = kmeans.labels_
Bracket match found on line: 307
Python 2
Tab Width: 8
Ln 307, Col 35
INS


Activities
Text Editor
Feb 17 18:13
Open
*la4_p2.py
~/Desktop
Save
# Deploy PCA
pca = PCA(n_components=2)
X_r = pca.fit_transform(X)
# Create matplotlib graphs
plt.figure(figsize=(12, 8))
for i in range(5):
plt.scatter(X_r[labels == i, 0], X_r[labels == i, 1], label=f'Cluster {i}')
plt.legend()
plt.title('K-Means Clustering')
plt.xlabel('PCA Feature 1')
plt.ylabel('PCA Feature 2')
plt.show()
# # Export csv
clustered_data.to_csv('clustered_data.csv', index=False)
#return X_r
def input_thread(interval_minutes):
global kmeans,model, X_r
try:
while True:
user_input = input().strip()
if user_input.lower() == "quit":
if connection.is_connected():
cursor.close()
connection.close()
os._exit(0)
else:
# Assume user input is a message or keywords
# Preprocess the input text
#X_r = display_clusters()
labels = kmeans.labels_
clustered_data = pd.read_csv('clustered_data.csv')
Python 2
Tab Width: 8
Ln 307, Col 35
INS

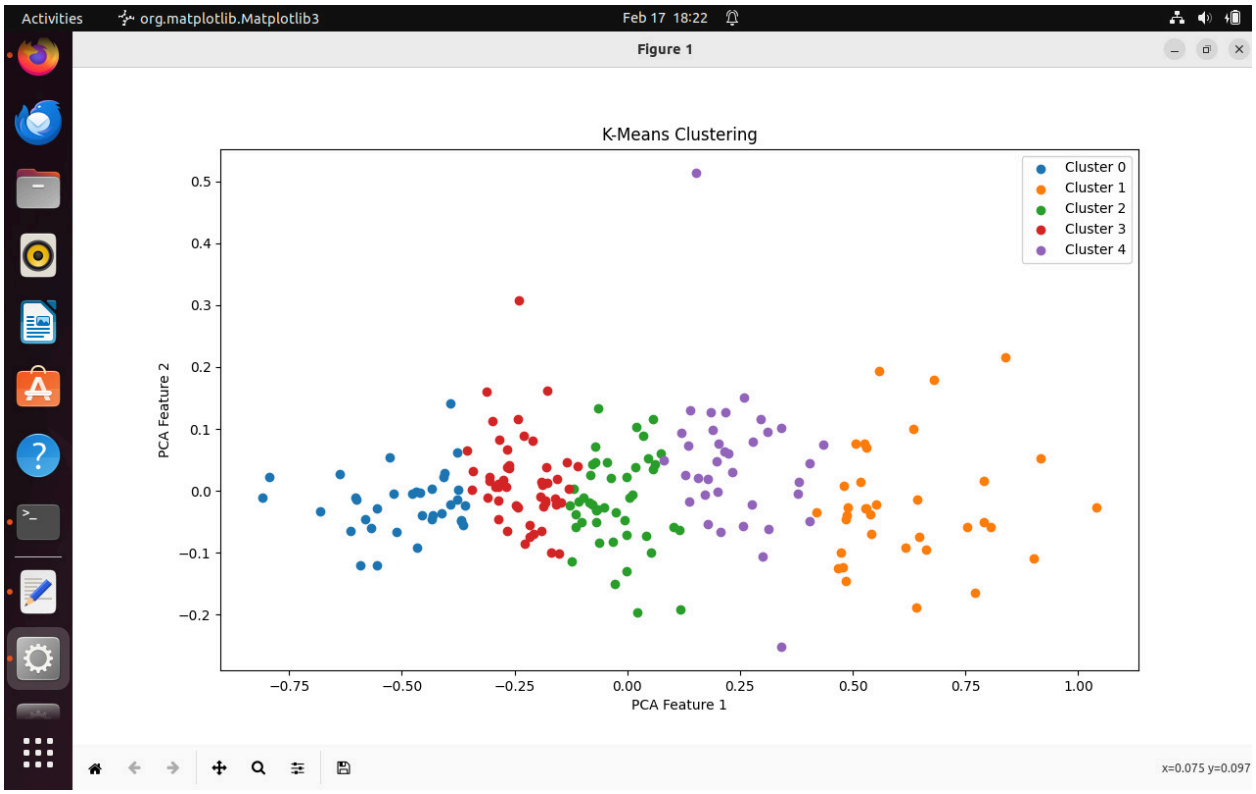


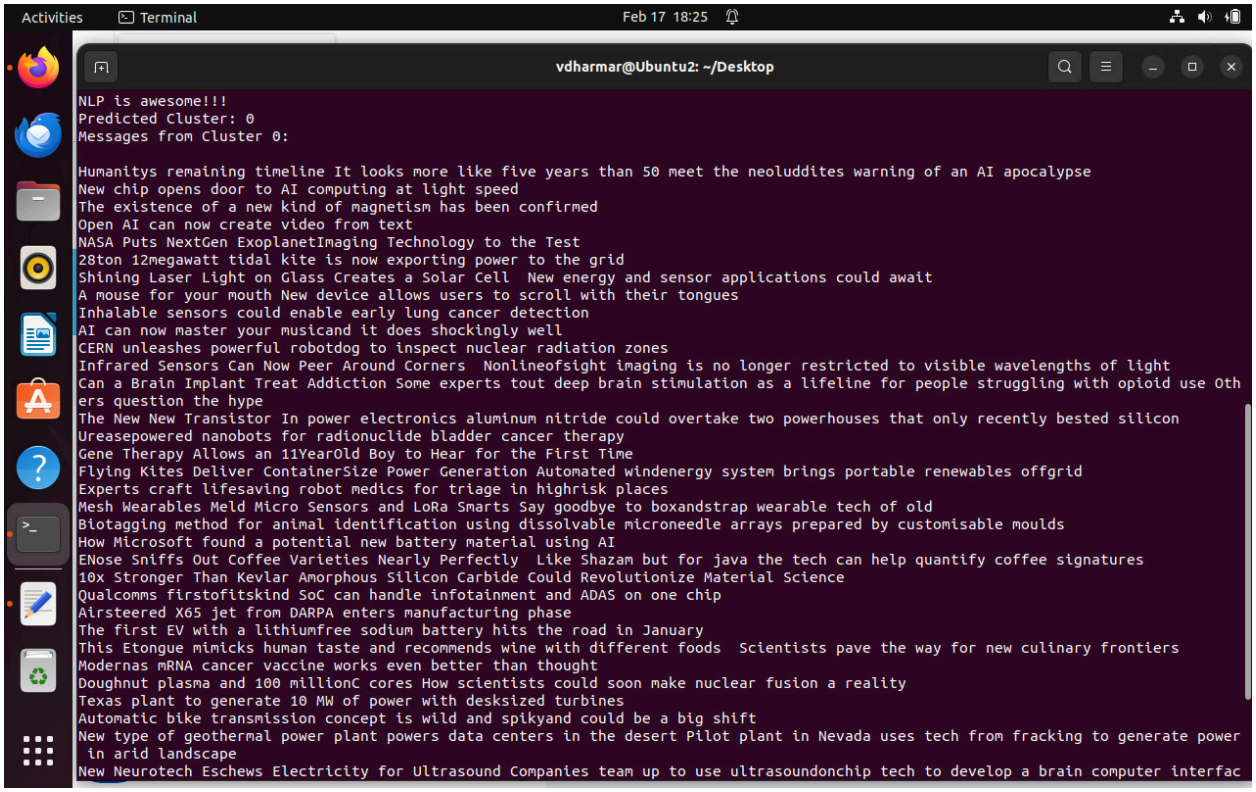


The Linux terminal output of the above script is shown below,

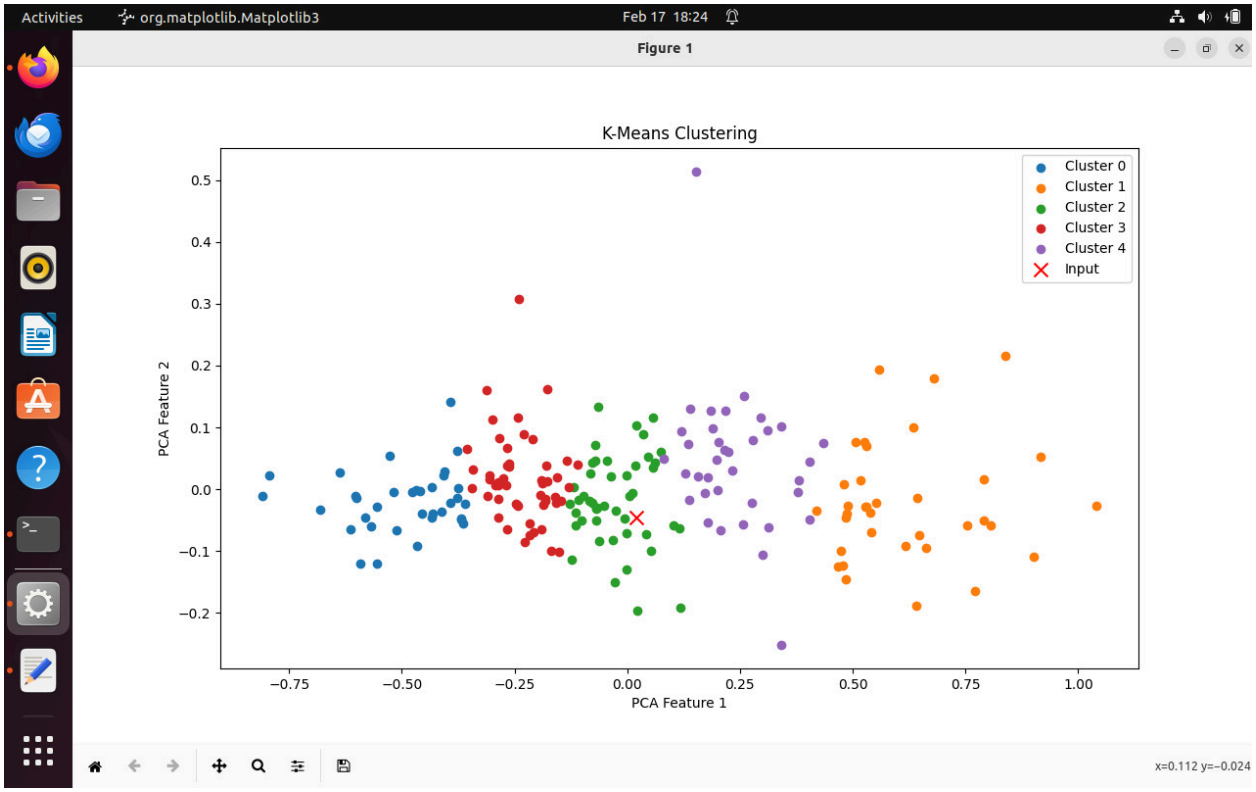


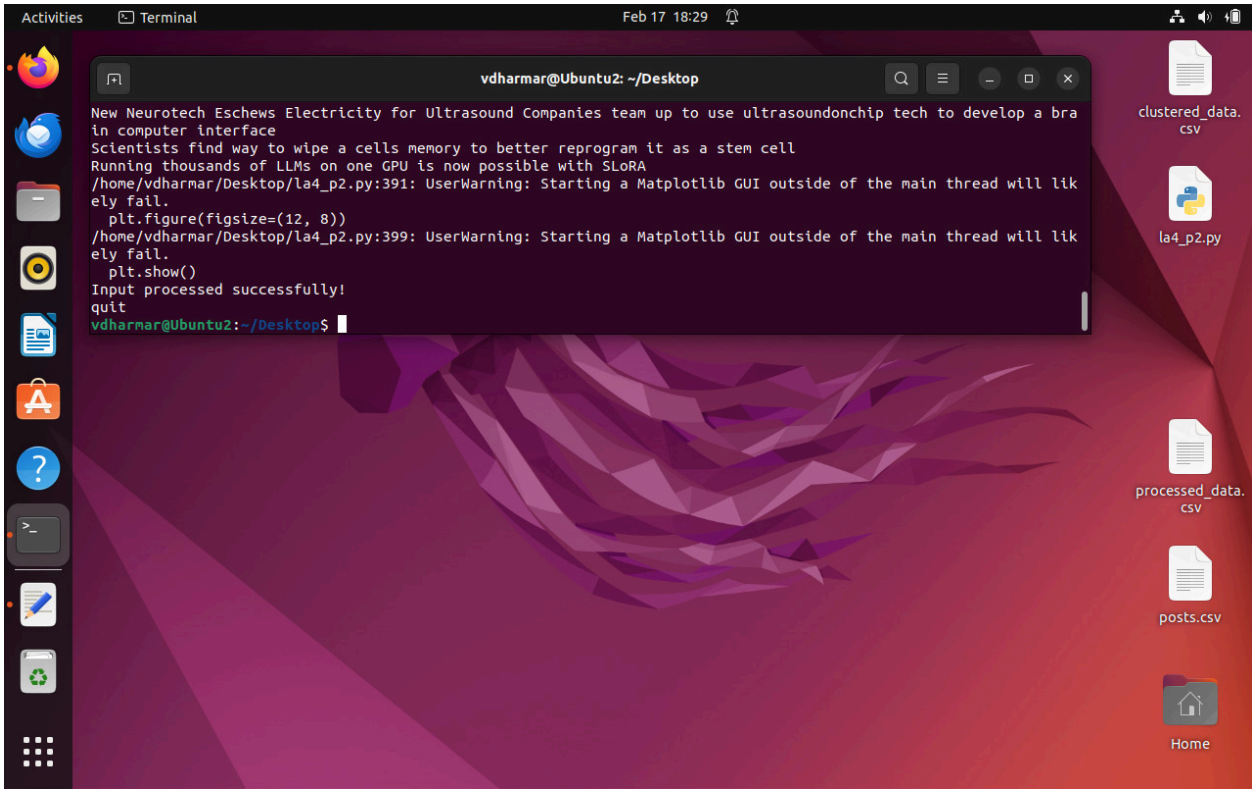
The below screenshot shows the graphical representation of the Clusters,



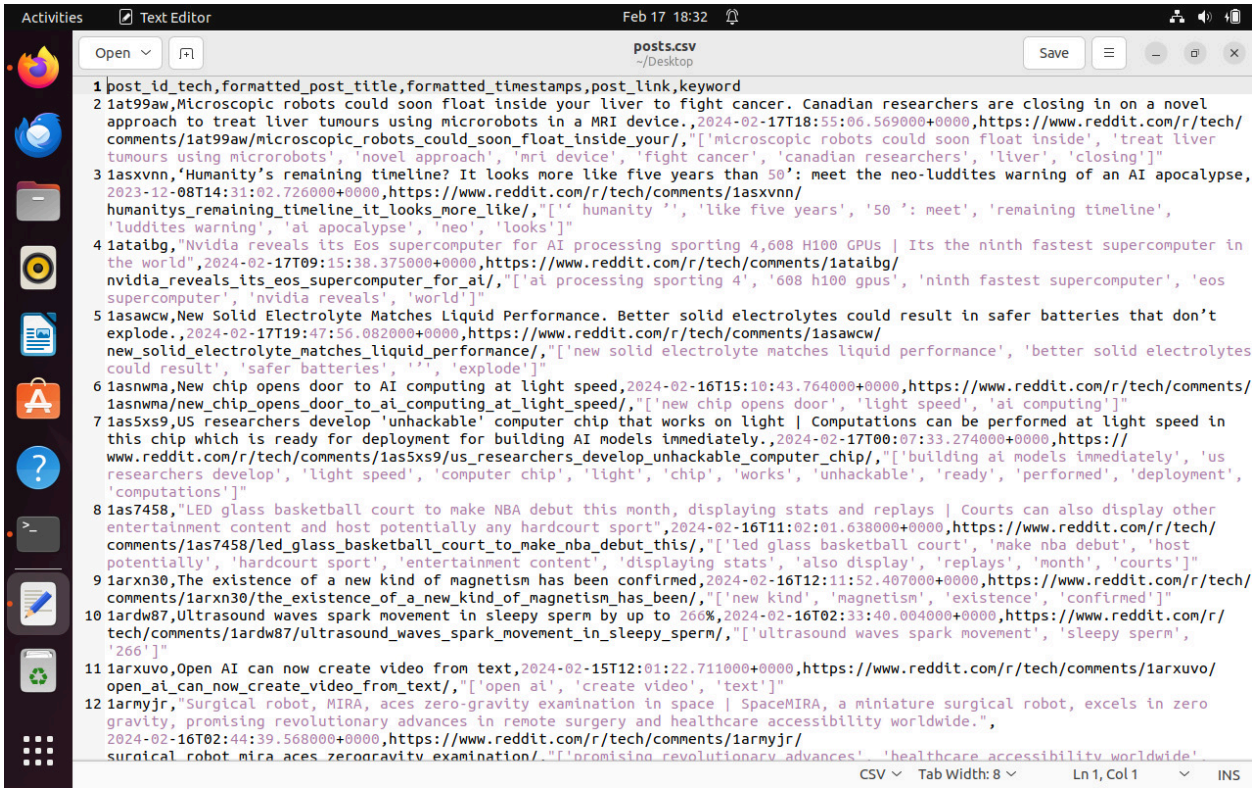


The below screenshot screenshot shows the graphical representation of the clusters and the position of the input,

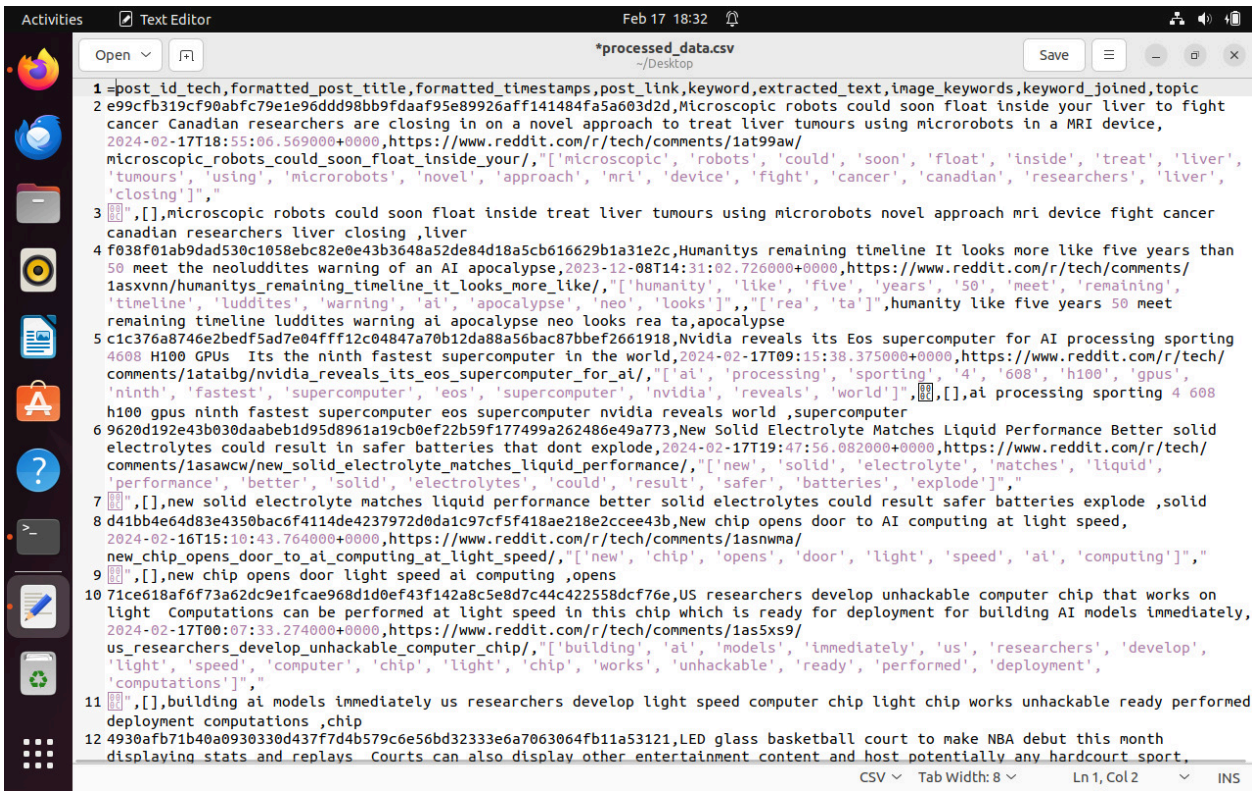




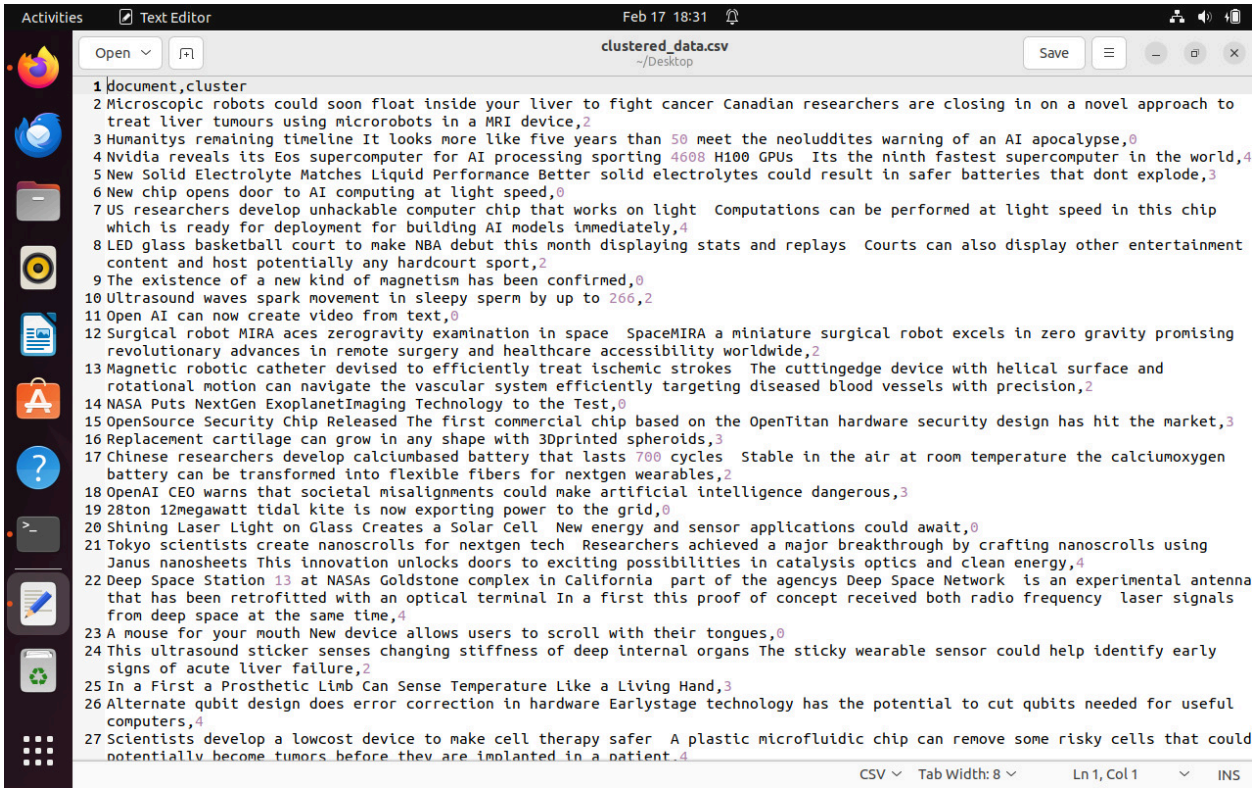
The screenshots below show the output of the CSVs generated,



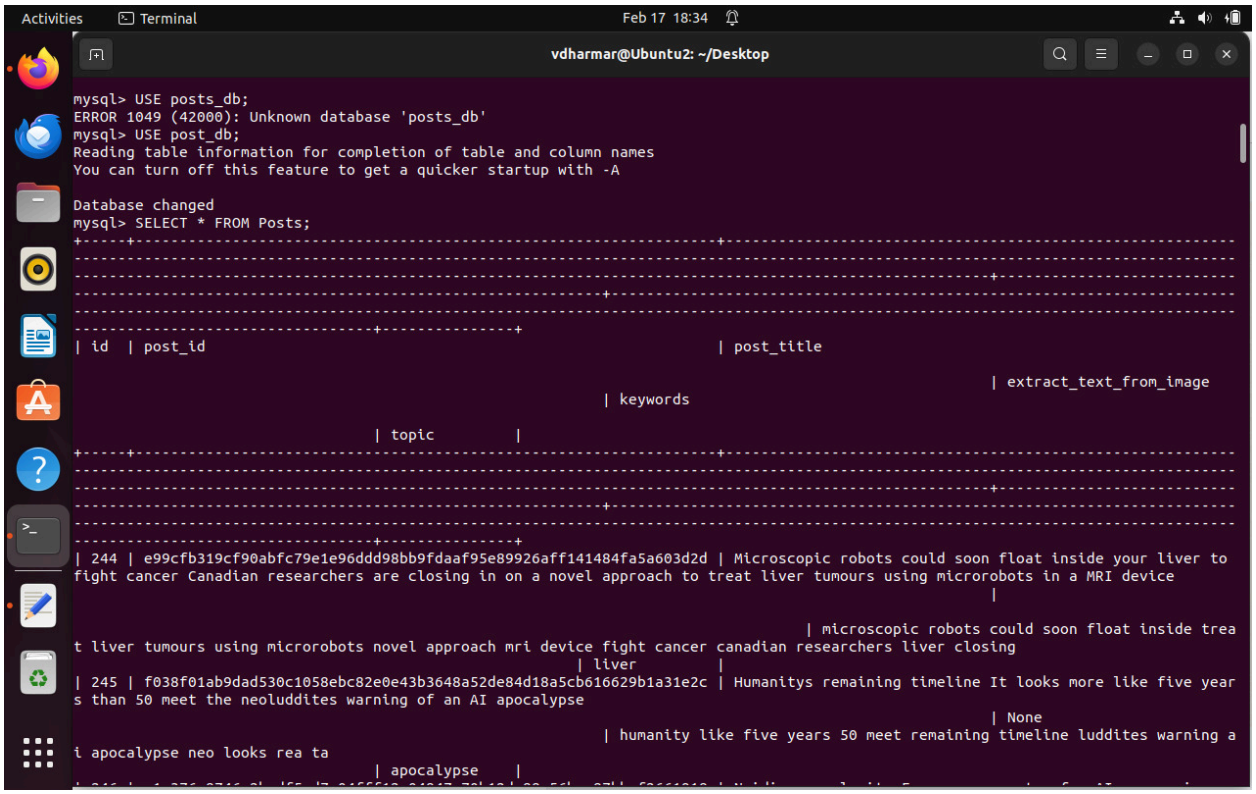

```
post_id_tech,formatted_post_title,formatted_timestamps,post_link,keyword
1at99aw,Microscopic robots could soon float inside your liver to fight cancer. Canadian researchers are closing in on a novel approach to treat liver tumours using microrobots in a MRI device.,2024-02-17T18:55:06.569000+0000,https://www.reddit.com/r/tech/comments/1at99aw/microscopic_robots_could_soon_float_inside_your/,"['microscopic robots could soon float inside', 'treat liver tumours using microrobots', 'novel approach', 'mri device', 'fight cancer', 'canadian researchers', 'liver', 'closing']"
1asxvnn,‘Humanity’s remaining timeline? It looks more like five years than 50’: meet the neo-luddites warning of an AI apocalypse,2023-12-08T14:31:02.726000+0000,https://www.reddit.com/r/tech/comments/1asxvnn/humanitys_remaining_timeline_it_looks_more_like/,"['‘ humanity ’', 'like five years', '50 ’: meet', 'remaining timeline', 'luddites warning', 'ai apocalypse', 'neo', 'looks']"
1ataibg,"Nvidia reveals its Eos supercomputer for AI processing sporting 4,608 H100 GPUs | Its the ninth fastest supercomputer in the world",2024-02-17T09:15:38.375000+0000,https://www.reddit.com/r/tech/comments/1ataibg/nvidia_reveals_its_eos_supercomputer_for_ai/,"['ai processing sporting 4', '608 h100 gpus', 'ninth fastest supercomputer', 'eos supercomputer', 'nvidia reveals', 'world']"
1asawcw,New Solid Electrolyte Matches Liquid Performance. Better solid electrolytes could result in safer batteries that don’t explode.,2024-02-17T19:47:56.082000+0000,https://www.reddit.com/r/tech/comments/1asawcw/new_solid_electrolyte_matches_liquid_performance/,"['new solid electrolyte matches liquid performance', 'better solid electrolytes could result', 'safer batteries', '’', 'explode']"
1asnwma,New chip opens door to AI computing at light speed,2024-02-16T15:10:43.764000+0000,https://www.reddit.com/r/tech/comments/1asnwma/new_chip_opens_door_to_ai_computing_at_light_speed/,"['new chip opens door', 'light speed', 'ai computing']"
1as5xs9,US researchers develop 'unhackable' computer chip that works on light | Computations can be performed at light speed in this chip which is ready for deployment for building AI models immediately.,2024-02-17T00:07:33.274000+0000,https://www.reddit.com/r/tech/comments/1as5xs9/us_researchers_develop_unhackable_computer_chip/,"['building ai models immediately', 'us researchers develop', 'light speed', 'computer chip', 'light', 'chip', 'works', 'unhackable', 'ready', 'performed', 'deployment', 'computations']"
1as7458,"LED glass basketball court to make NBA debut this month, displaying stats and replays | Courts can also display other entertainment content and host potentially any hardcourt sport",2024-02-16T11:02:01.638000+0000,https://www.reddit.com/r/tech/comments/1as7458/led_glass_basketball_court_to_make_nba_debut_this/,"['led glass basketball court', 'make nba debut', 'host potentially', 'hardcourt sport', 'entertainment content', 'displaying stats', 'also display', 'replays', 'month', 'courts']"
1arxn30,The existence of a new kind of magnetism has been confirmed,2024-02-16T12:11:52.407000+0000,https://www.reddit.com/r/tech/comments/1arxn30/the_existence_of_a_new_kind_of_magnetism_has_been/,"['new kind', 'magnetism', 'existence', 'confirmed']"
1ardw87,Ultrasound waves spark movement in sleepy sperm by up to 266%,2024-02-16T02:33:40.004000+0000,https://www.reddit.com/r/tech/comments/1ardw87/ultrasound_waves_spark_movement_in_sleepy_sperm/,"['ultrasound waves spark movement', 'sleepy sperm', '266']"
1arxuvo,Open AI can now create video from text,2024-02-15T12:01:22.711000+0000,https://www.reddit.com/r/tech/comments/1arxuvo/open_ai_can_now_create_video_from_text/,"['open ai', 'create video', 'text']"
1armyjr,"Surgical robot, MIRA, aces zero-gravity examination in space | SpaceMIRA, a miniature surgical robot, excels in zero gravity, promising revolutionary advances in remote surgery and healthcare accessibility worldwide.",2024-02-16T02:44:39.568000+0000,https://www.reddit.com/r/tech/comments/1armyjr/surgical_robot_mira_aces_zerogravity_examination/,"['promising revolutionary advances', 'healthcare accessibility worldwide',
```

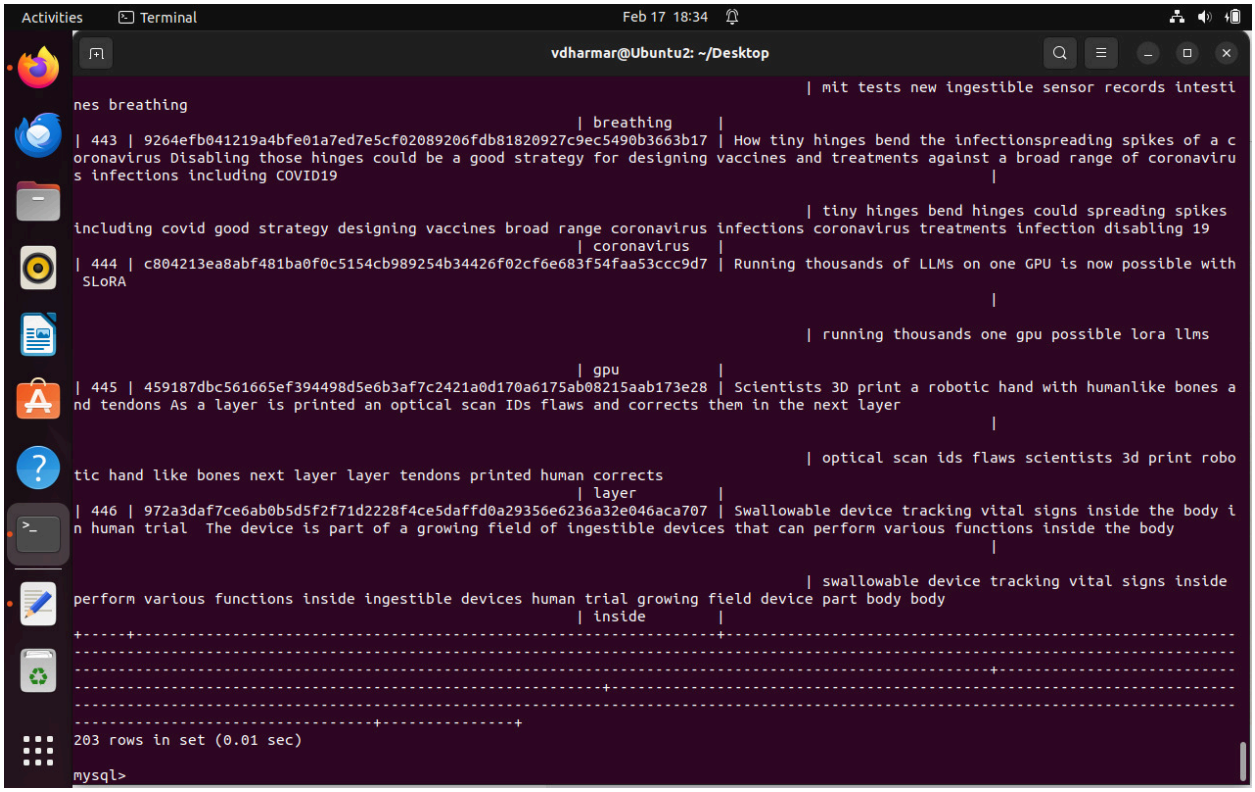



```
=post_id_tech,formatted_post_title,formatted_timestamps,post_link,keyword,extracted_text,image_keywords,keyword_joined,topic
e99cfb319cf90abfc79e1e96ddd98bb9fdaaf95e89926aff141484fa5a603d2d,Microscopic robots could soon float inside your liver to fight cancer Canadian researchers are closing in on a novel approach to treat liver tumours using microrobots in a MRI device,2024-02-17T18:55:06.569000+0000,https://www.reddit.com/r/tech/comments/1at99aw/microscopic_robots_could_soon_float_inside_your/,"['microscopic', 'robots', 'could', 'soon', 'float', 'inside', 'treat', 'liver', 'tumours', 'using', 'microrobots', 'novel', 'approach', 'mri', 'device', 'fight', 'cancer', 'canadian', 'researchers', 'liver', 'closing']","
",[],microscopic robots could soon float inside treat liver tumours using microrobots novel approach mri device fight cancer canadian researchers liver closing ,liver
f038f01ab9dad530c1058ebc82e0e43b3648a52de84d18a5cb616629b1a31e2c,Humanitys remaining timeline It looks more like five years than 50 meet the neoluddites warning of an AI apocalypse,2023-12-08T14:31:02.726000+0000,https://www.reddit.com/r/tech/comments/1asxvnn/humanitys_remaining_timeline_it_looks_more_like/,"['humanity', 'like', 'five', 'years', '50', 'meet', 'remaining', 'timeline', 'luddites', 'warning', 'ai', 'apocalypse', 'neo', 'looks']",,"['rea', 'ta']",humanity like five years 50 meet remaining timeline luddites warning ai apocalypse neo looks rea ta,apocalypse
c1c376a8746e2bedf5ad7e04fff12c04847a70b12da88a56bac87bbef2661918,Nvidia reveals its Eos supercomputer for AI processing sporting 4608 H100 GPUs  Its the ninth fastest supercomputer in the world,2024-02-17T09:15:38.375000+0000,https://www.reddit.com/r/tech/comments/1ataibg/nvidia_reveals_its_eos_supercomputer_for_ai/,"['ai', 'processing', 'sporting', '4', '608', 'h100', 'gpus', 'ninth', 'fastest', 'supercomputer', 'eos', 'supercomputer', 'nvidia', 'reveals', 'world']","
",[],ai processing sporting 4 608 h100 gpus ninth fastest supercomputer eos supercomputer nvidia reveals world ,supercomputer
9620d192e43b030daabeb1d95d8961a19cb0ef22b59f177499a262486e49a773,New Solid Electrolyte Matches Liquid Performance Better solid electrolytes could result in safer batteries that dont explode,2024-02-17T19:47:56.082000+0000,https://www.reddit.com/r/tech/comments/1asawcw/new_solid_electrolyte_matches_liquid_performance/,"['new', 'solid', 'electrolyte', 'matches', 'liquid', 'performance', 'better', 'solid', 'electrolytes', 'could', 'result', 'safer', 'batteries', 'explode']","
",[],new solid electrolyte matches liquid performance better solid electrolytes could result safer batteries explode ,solid
d41bb4e64d83e4350bac6f4114de4237972d0da1c97cf5f418ae218e2ccee43b,New chip opens door to AI computing at light speed,2024-02-16T15:10:43.764000+0000,https://www.reddit.com/r/tech/comments/1asnwma/new_chip_opens_door_to_ai_computing_at_light_speed/,"['new', 'chip', 'opens', 'door', 'light', 'speed', 'ai', 'computing']","
",[],new chip opens door light speed ai computing ,opens
71ce618af6f73a62dc9e1fcae968d1d0ef43f142a8c5e8d7c44c422558dcf76e,US researchers develop unhackable computer chip that works on light  Computations can be performed at light speed in this chip which is ready for deployment for building AI models immediately,2024-02-17T00:07:33.274000+0000,https://www.reddit.com/r/tech/comments/1as5xs9/us_researchers_develop_unhackable_computer_chip/,"['building', 'ai', 'models', 'immediately', 'us', 'researchers', 'develop', 'light', 'speed', 'computer', 'chip', 'light', 'chip', 'works', 'unhackable', 'ready', 'performed', 'deployment', 'computations']","
",[],building ai models immediately us researchers develop light speed computer chip light chip works unhackable ready performed deployment computations ,chip
4930afb71b40a0930330d437f7d4b579c6e56bd32333e6a7063064fb11a53121,LED glass basketball court to make NBA debut this month displaying stats and replays  Courts can also display other entertainment content and host potentially any hardcourt sport,
```



The below screenshots show the output records stored in the database,




Activities
Terminal
Feb 17 18:34
vdharmar@Ubuntu2: ~/Desktop
| mit tests new ingestible sensor records intesti
nes breathing
| breathing |
| 443 | 9264efb041219a4bfe01a7ed7e5cf02089206fdb81820927c9ec5490b3663b17 | How tiny hinges bend the infectionspreading spikes of a c
oronavirus Disabling those hinges could be a good strategy for designing vaccines and treatments against a broad range of coronaviru
s infections including COVID19 |
| tiny hinges bend hinges could spreading spikes
including covid good strategy designing vaccines broad range coronavirus infections coronavirus treatments infection disabling 19
| coronavirus |
| 444 | c804213ea8abf481ba0f0c5154cb989254b34426f02cf6e683f54faa53ccc9d7 | Running thousands of LLMs on one GPU is now possible with
SLoRA |
| running thousands one gpu possible lora llms
| gpu |
| 445 | 459187dbc561665ef394498d5e6b3af7c2421a0d170a6175ab08215aab173e28 | Scientists 3D print a robotic hand with humanlike bones a
nd tendons As a layer is printed an optical scan IDs flaws and corrects them in the next layer |
| optical scan ids flaws scientists 3d print robo
tic hand like bones next layer layer tendons printed human corrects
| layer |
| 446 | 972a3daf7ce6ab0b5d5f2f71d2228f4ce5daffd0a29356e6236a32e046aca707 | Swallowable device tracking vital signs inside the body i
n human trial The device is part of a growing field of ingestible devices that can perform various functions inside the body |
| swallowable device tracking vital signs inside
perform various functions inside ingestible devices human trial growing field device part body body
| inside |
203 rows in set (0.01 sec)
mysql>