به نام خدا

دانشگاه اصفهان
دانشکده مهندسی کامپیوتر

پروژه پایانی درس شبکه های اجتماعی

استاد درس: استاد خانیکی
دستیاران: فریبا عزیزیان

سرتیم: ملیکا آقاجانیان صباغ ۹۹۳۶۲۳۰۰۴
اعضا: مهدیس فتحی ۹۹۳۶۱۳۰۴۹

melika-aghajanian/Analyze-Twitter-DataSet-about-Chat-GPT- (github.com)

فهرست

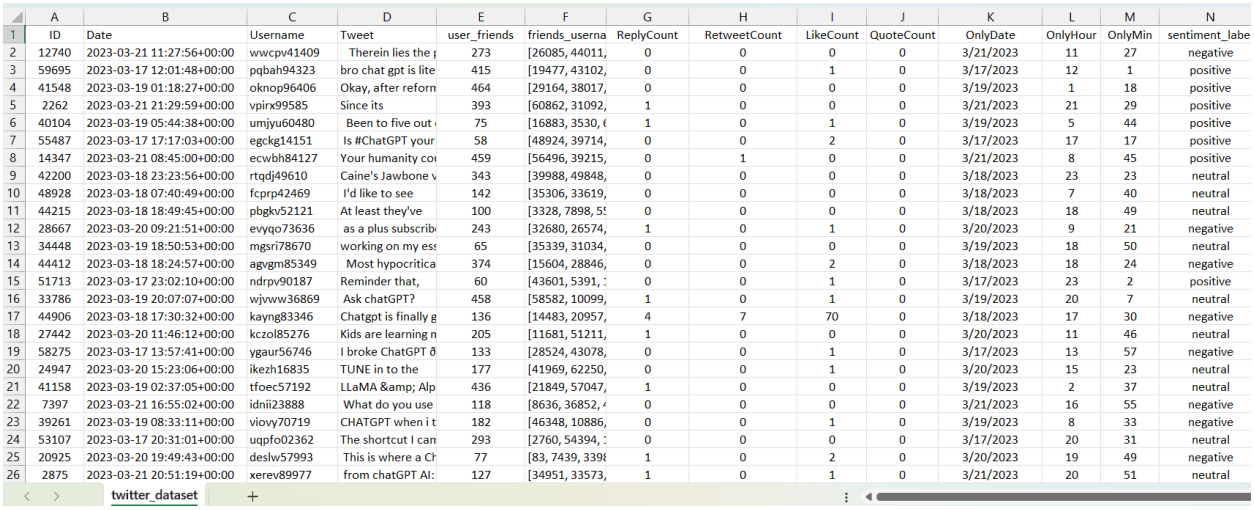
گام اول

دیتا ست استفاده شده در این پروژه مرتبط اکانت های توییتری می باشد که توییت هایی درباره هوش مصنوعی chat GPT نوشته و به اشتراک گذاشته اند.
این دیتا ست دارای ۵۰۰ نود و ارتباط بین انها follower و following می باشد.

با استفاده از قطعه کد زیر از دیتاست اصلی ۵۰۰ گره را استخراج کردیم.

```python
import pandas as pd

def reduce_dataset(input_file, output_file, target_size=500):
    # Load the entire dataset
    df = pd.read_csv(input_file)

    # Sample a subset of the dataset
    sampled_df = df.sample(n=min(target_size, len(df)))

    # Save the reduced dataset to a new CSV file
    sampled_df.to_csv(output_file, index=False)

if __name__ == "__main__":
    # Load the training data
    input_file = 'twitter_data.csv'
    output_file = 'twitter_dataset.csv'
    reduce_dataset(input_file, output_file)
```

این دیتاست دارای ستون های زیر می باشد و کاربرد هریک اینگونه تعریف میشود:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ID	Date	Username	Tweet	user_friends	friends_userna	ReplyCount	RetweetCount	LikeCount	QuoteCount	OnlyDate	OnlyHour	OnlyMin	sentiment_labe
2	12740	2023-03-21 11:27:56+00:00	wwcpv41409	Therein lies the p	273	[26085, 44011,	0	0	0	0	3/21/2023	11	27	negative
3	59695	2023-03-17 12:01:48+00:00	pqbah94323	bro chat gpt is lite	415	[19477, 43102,	0	0	1	0	3/17/2023	12	1	positive
4	41548	2023-03-19 01:18:27+00:00	oknop96406	Okay, after reform	464	[29164, 38017,	0	0	0	0	3/19/2023	1	18	positive
5	2262	2023-03-21 21:29:59+00:00	vpirx99585	Since its	393	[60862, 31092,	1	0	0	0	3/21/2023	21	29	positive
6	40104	2023-03-19 05:44:38+00:00	umjyu60480	Been to five out	75	[16883, 3530, (	1	0	1	0	3/19/2023	5	44	positive
7	55487	2023-03-17 17:17:03+00:00	egckg14151	Is #ChatGPT your	58	[48924, 39714,	0	0	2	0	3/17/2023	17	17	positive
8	14347	2023-03-21 08:45:00+00:00	ecwbh84127	Your humanity cou	459	[56496, 39215,	0	1	0	0	3/21/2023	8	45	positive
9	42200	2023-03-18 23:23:56+00:00	rtqdj49610	Caine's Jawbone v	343	[39988, 49848,	0	0	0	0	3/18/2023	23	23	neutral
10	48928	2023-03-18 07:40:49+00:00	fcprp42469	I'd like to see	142	[35306, 33619,	0	0	0	0	3/18/2023	7	40	neutral
11	44215	2023-03-18 18:49:45+00:00	pbgkv52121	At least they've	100	[3328, 7898, 55	0	0	0	0	3/18/2023	18	49	neutral
12	28667	2023-03-20 09:21:51+00:00	evyqo73636	as a plus subscrib	243	[32680, 26574,	1	0	1	0	3/20/2023	9	21	negative
13	34448	2023-03-19 18:50:53+00:00	mgsri78670	working on my ess	65	[35339, 31034,	0	0	0	0	3/19/2023	18	50	neutral
14	44412	2023-03-18 18:24:57+00:00	agvgm85349	Most hypocritica	374	[15604, 28846,	0	0	2	0	3/18/2023	18	24	negative
15	51713	2023-03-17 23:02:10+00:00	ndrpv90187	Reminder that,	60	[43601, 5391, :	0	0	1	0	3/17/2023	23	2	positive
16	33786	2023-03-19 20:07:07+00:00	wjvww36869	Ask chatGPT?	458	[58582, 10099,	1	0	1	0	3/19/2023	20	7	neutral
17	44906	2023-03-18 17:30:32+00:00	kayng83346	Chatgpt is finally g	136	[14483, 20957,	4	7	70	0	3/18/2023	17	30	negative
18	27442	2023-03-20 11:46:12+00:00	kczol85276	Kids are learning n	205	[11681, 51211,	1	0	0	0	3/20/2023	11	46	neutral
19	58275	2023-03-17 13:57:41+00:00	ygaur56746	I broke ChatGPT ð	133	[28524, 43078,	0	0	1	0	3/17/2023	13	57	negative
20	24947	2023-03-20 15:23:06+00:00	ikezh16835	TUNE in to the	177	[41969, 62250,	0	0	1	0	3/20/2023	15	23	neutral
21	41158	2023-03-19 02:37:05+00:00	tfoec57192	LLaMA & Alp	436	[21849, 57047,	1	0	0	0	3/19/2023	2	37	neutral
22	7397	2023-03-21 16:55:02+00:00	idnii23888	What do you use	118	[8636, 36852, (	0	0	0	0	3/21/2023	16	55	negative
23	39261	2023-03-19 08:33:11+00:00	viovy70719	CHATGPT when i t	182	[46348, 10886,	0	0	1	0	3/19/2023	8	33	negative
24	53107	2023-03-17 20:31:01+00:00	uqpfo02362	The shortcut I can	293	[2760, 54394, :	0	0	0	0	3/17/2023	20	31	neutral
25	20925	2023-03-20 19:49:43+00:00	deslw57993	This is where a Ch	77	[83, 7439, 3398	1	0	2	0	3/20/2023	19	49	negative
26	2875	2023-03-21 20:51:19+00:00	xerev89977	from chatGPT AI:	127	[34951, 33573,	1	0	1	0	3/21/2023	20	51	neutral

twitter_dataset

- Id: id اختصاصی برای هر اکانت تویتر در دیتاست.
- Username: ایدی اکانتی که توییت مربوطه را منتشر کرده است.
- Tweet: متن اصلی توییت منتشر شده.
- User friends: تعداد فالورهای اکانتی که توییت توسط ان منتشر شده است.
- Friends username: لیست فالورهای اکانتی که توییت توسط او منتشر شده است.
- Replycount: تعداد ریپلای ها روی هر توییت زده شده.
- Retweetcount: تعداد retweet های زده شده از روی هر توییت.
- Likecount: تعداد لایک های هر توییت.
- Quotcount: تعداد quot های زده شده از روی هر توییت.
- Date: مجموعه زمانی اعم از تاریخ، ساعت، دقیقه و ثانیه انتشار توییت/
- only date: تاریخی که توییت منتشر شده است.
- Only hour: ساعت انتشار توییت.
- Only min: دقیقه انتشار توییت.
- Sentimeter_lable: تحلیل توییت از نظر مثبت بودن – منفی بودن و خنثی بودن.

گام دوم

یرای محاسبه معیار های اهمیت و مرکزیت دیتاست را با استفاده از قطعه کد زیر به گراف تبدیل کرده و ارتباطات هر نود را با سایر نودهای دیگر به آن اضافه کردیم.

سپس با استفاده از قطعه کد زیر معیارهای مرکزیت و اهمیت زیر را برای ۵ نود محاسبه کردیم.

- Degree
- Between
- Closeness
- Eigenvector
- Pagerank

```
G = nx.Graph()

for index, row in df.iterrows():
    ID = row['ID']
    friends_str = row['friends_username']
    friends_list = ast.literal_eval(friends_str)

    for friend in friends_list:
        if not G.has_node(friend):
            G.add_node(friend)
        G.add_edge(ID, friend)

# Calculate centrality measures for a subset of nodes
subset_nodes = random.sample(list(G.nodes()), min(500, len(G)))  # Select 500
random nodes or all nodes if the graph is smaller
subset_graph = G.subgraph(subset_nodes)

# Degree centrality
degree_centrality = nx.degree_centrality(subset_graph)
top_degree_nodes = sorted(degree_centrality, key=degree_centrality.get,
reverse=True)[:5]
print("Nodes with the highest degree centrality: ", top_degree_nodes)

# Betweenness centrality
betweenness_centrality = nx.betweenness_centrality(subset_graph)
```

```
top_betweenness_nodes = sorted(betweenness_centrality,
key=betweenness_centrality.get, reverse=True)[:5]
print("Nodes with the highest betweenness centrality: ", top_betweenness_nodes)

# Closeness centrality
closeness_centrality = nx.closeness_centrality(subset_graph)
top_closeness_nodes = sorted(closeness_centrality, key=closeness_centrality.get,
reverse=True)[:5]
print("Nodes with the highest closeness centrality: ", top_closeness_nodes)

# Eigenvector centrality
eigenvector_centrality = nx.eigenvector_centrality(subset_graph)
top_eigenvector_nodes = sorted(eigenvector_centrality,
key=eigenvector_centrality.get, reverse=True)[:5]
print("Nodes with the highest eigenvector centrality: ", top_eigenvector_nodes)

# PageRank
pagerank = nx.pagerank(subset_graph)
top_pagerank_nodes = sorted(pagerank, key=pagerank.get, reverse=True)[:5]
print("Nodes with the highest PageRank: ", top_pagerank_nodes)
```

خروجی قطعه کد بالا:

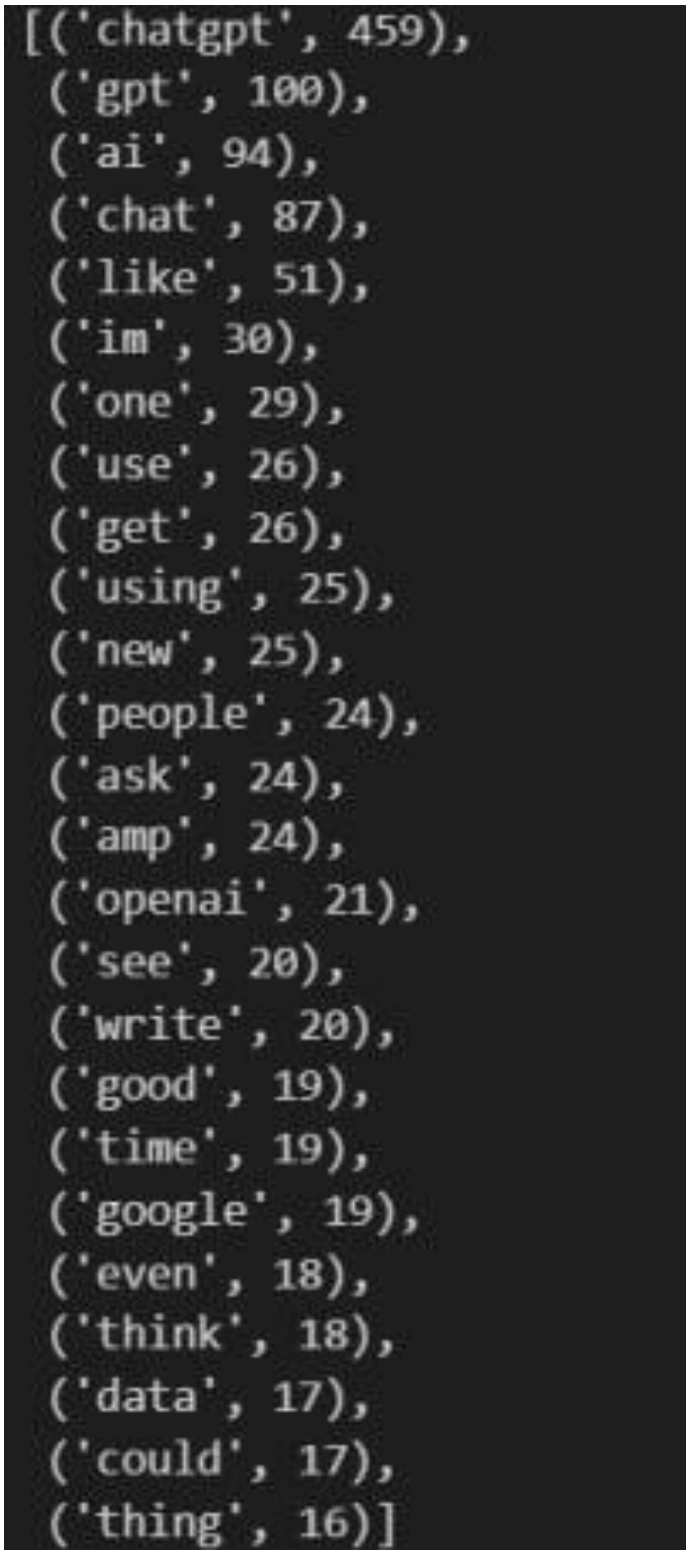
```
Nodes with the highest degree centrality:  [22532, 20484, 61445, 30734, 4114]
Nodes with the highest betweenness centrality:  [22532, 20484, 61445, 30734, 4114]
Nodes with the highest closeness centrality:  [22532, 20484, 61445, 30734, 4114]
Nodes with the highest eigenvector centrality:  [22532, 20484, 61445, 30734, 4114]
Nodes with the highest PageRank:  [22532, 20484, 61445, 30734, 4114]
```

گام سوم

با استفاده از قطعه کد زیر تمامی توییت هایی را که در ستون Tweet در دیتاست ذخیره کرده بودیم را بررسی کردیم و ۲۵ کلمه کلیدی و اساسی را بین تمامی توییت ها پیدا کردیم.

```
# Preprocessing function to remove special characters and convert to lowercase
def preprocess_text(text):
    cleaned_text = re.sub(r"http\S+|[^a-zA-Z\s]", "", text)
    cleaned_text = cleaned_text.lower()
    return cleaned_text

# Tokenization function
def tokenize_text(text):
    tokens = word_tokenize(text)
    return tokens

# Apply preprocessing and tokenization to each tweet
df["cleaned_tweet"] = df["Tweet"].apply(preprocess_text)
df["tokens"] = df["cleaned_tweet"].apply(tokenize_text)

# Concatenate all tokens into a single list
all_tokens = [token for tweet_tokens in df["tokens"] for token in tweet_tokens]

# Remove stopwords from the list of tokens
stop_words = set(stopwords.words('english'))
filtered_tokens = [token for token in all_tokens if token.lower() not in
stop_words]

# Calculate word frequencies
fdist = FreqDist(filtered_tokens)

# Get the 25 most common words
most_common_words = fdist.most_common(25)

most_common_words
```

خروجی قطعه کد بالا



```
[('chatgpt', 459),
 ('gpt', 100),
 ('ai', 94),
 ('chat', 87),
 ('like', 51),
 ('im', 30),
 ('one', 29),
 ('use', 26),
 ('get', 26),
 ('using', 25),
 ('new', 25),
 ('people', 24),
 ('ask', 24),
 ('amp', 24),
 ('openai', 21),
 ('see', 20),
 ('write', 20),
 ('good', 19),
 ('time', 19),
 ('google', 19),
 ('even', 18),
 ('think', 18),
 ('data', 17),
 ('could', 17),
 ('thing', 16)]
```

با استفاده از فطعه کد زیر بررسی کردیم در بین تمامی توییت ها چه تعداد از کلمات بار مثبت، چه تعداد بار منفی و چه تعداد خنثی هستند:

```
# Download the required resources (run only once)
nltk.download('vader_lexicon')

# Create an instance of the SentimentIntensityAnalyzer class
sia = SentimentIntensityAnalyzer()

# Function to calculate the sentiment label for a word
def get_sentiment_label(word):
    sentiment_scores = sia.polarity_scores(word)
    compound_score = sentiment_scores['compound']
    if compound_score > 0:
        return "positive"
    elif compound_score < 0:
        return "negative"
    else:
        return "neutral"

# Apply the get_sentiment_label function to each word in the dataset
df["sentiment_label"] = df["tokens"].apply(lambda tokens:
[get_sentiment_label(word) for word in tokens])

# Count the number of negative, positive, and neutral words
negative_count = sum(df["sentiment_label"].apply(lambda labels:
labels.count("negative")))
positive_count = sum(df["sentiment_label"].apply(lambda labels:
labels.count("positive")))
neutral_count = sum(df["sentiment_label"].apply(lambda labels:
labels.count("neutral")))

print("Negative words:", negative_count)
print("Positive words:", positive_count)
print("Neutral words:", neutral_count)
```

خروجی قطعه کد بالا

```
word semantic count in all of the tweets:
Negative words: 215
Positive words: 542
Neutral words: 9532
```

سپس با استفاده از قطعه کد زیر به بررسی بار مثبت، منفی و خنثی کلمات کلیدی و مهم توییت ها که در مرحله قبلی پیدا کردیم پرداختیم:

```
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer

# Download the required resources (run only once)
nltk.download('vader_lexicon')

# Create an instance of the SentimentIntensityAnalyzer class
sia = SentimentIntensityAnalyzer()

# Function to calculate the sentiment label for a word
def get_sentiment_label(word):
    sentiment_scores = sia.polarity_scores(word)
    compound_score = sentiment_scores['compound']
    if compound_score > 0:
        return "positive"
    elif compound_score < 0:
        return "negative"
    else:
        return "neutral"

# Apply the get_sentiment_label function to each word in the most common words
most_common_sentiments = []
for word, frequency in most_common_words:
    sentiment_label = get_sentiment_label(word)
    most_common_sentiments.append((word, frequency, sentiment_label))
```

```
# Count the number of negative, positive, and neutral words among the most common words
negative_count = sum(1 for _, _, sentiment in most_common_sentiments if sentiment == "negative")
positive_count = sum(1 for _, _, sentiment in most_common_sentiments if sentiment == "positive")
neutral_count = sum(1 for _, _, sentiment in most_common_sentiments if sentiment == "neutral")

# Print the words in each sentiment group
negative_words = [word for word, _, sentiment in most_common_sentiments if sentiment == "negative"]
positive_words = [word for word, _, sentiment in most_common_sentiments if sentiment == "positive"]
neutral_words = [word for word, _, sentiment in most_common_sentiments if sentiment == "neutral"]

print("word semantic count in the most common words:")
print("Negative words:", negative_count)
print(negative_words)
print()
print("Positive words:", positive_count)
print(positive_words)
print()
print("Neutral words:", neutral_count)
print(neutral_words)
```

خروجی قطعه کد بالا

گام چهارم

برای مشخص کردن کاربران و میزان تاثیر گذاری ان ها بر کل شبکه از قطعه کد زیر استفاده کردیم.
برای محاسبه کاربران فعال در توییتر برای هر کاربر محتوای ستون های ,ReplyCount likeCount, QuotCount و retweetCount را محاسبه کردیم و این ستون جدید را به دیتا ست اضافه کردیم.

```
# Calculate the total engagement for each user
df['TotalEngagement'] = df['QuoteCount'] + df['LikeCount'] + df['RetweetCount'] +
df['ReplyCount']

# Sort the DataFrame by total engagement in descending order
df_sorted = df.sort_values(by='TotalEngagement', ascending=False)

# Filter the DataFrame for TotalEngagement greater than 0
active_users = df[df['TotalEngagement'] > 20]

# Extract the Username and ID columns into active_users
active_users = active_users[['Username', 'ID', 'TotalEngagement']]

print(active_users)
```

خروجی قطعه کد بالا

	Username	ID	TotalEngagement
15	kayng83346	44906	81
34	nfswv10740	23324	29
75	vxlju50938	25896	201
102	xdapa81984	1608	23
152	qyizz78556	39400	27
202	ivgbl54788	45970	54
203	szmnv56075	16395	21
234	tfpgz03850	24602	55
267	pnrqu00608	54532	206
341	oezep61822	29930	31
363	yjquz01947	56157	1316
368	gwbxx00775	51742	29
370	zduap99284	23643	21
379	nvrac18294	56661	544
382	obpnw19461	7212	34
447	ghshu68007	14821	21
478	mxmzm77440	17473	34
488	vnoec46240	33586	29
499	eihwf92101	38765	21

سپس برای محاسبه میزان تاثیرگذاری ۱۵ کاربر فعال و برتر شبکه میزان محبوبیت ان ها را که در مرحله قبلی به دست اورده بودیم را در تعداد فالورهای این کاربر ضرب کردیم.

```
# Calculate the content impression for each user
df_sorted['ContentImpression'] = df_sorted['TotalEngagement'] *
df_sorted['user_friends']

# Select the top 15 active users
top_users = df_sorted.head(15)

# Print the content impression of each user and the total content impression
print("Content Impression of Top 15 Active Users:")
print(top_users[['Username', 'ID' 'ContentImpression']])
```

خروجی قطعه کد بالا

```
Content Impression of Top 15 Active Users:
      Username     ID  ContentImpression
363  yjquz01947  56157              69748
379  nvrac18294  56661              17408
267  pnrqu00608  54532              47998
75   vxlju50938  25896              92259
15   kayng83346  44906              11016
234  tfpgz03850  24602               6380
202  ivgbl54788  45970               9936
478  mxmzm77440  17473               1326
382  obpnw19461   7212              11050
341  oezep61822  29930              14787
488  vnoec46240  33586                377
34   nfswv10740  23324               6467
368  gwbxx00775  51742                464
152  qyizz78556  39400              10989
102  xdapa81984   1608               4922
```

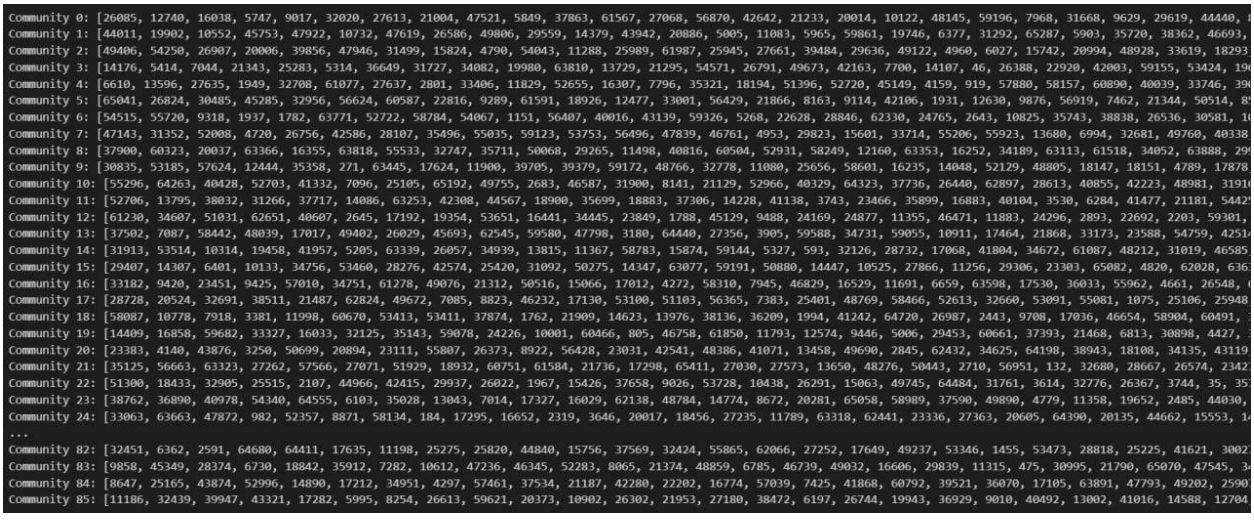
گام پنجم

برای محاسبه جامعه های فعال با استفاده از قطعه کد زیر گرافی را که از روی دیتاست رسم کرده بودیم، بررسی کردیم. سپس با استفاده از الگوریتم `Louvain` بر روی گراف اجرا کردیم. و گره ها را به انجمن‌ها`(communities)` تقسیم و در `partition` ذخیره می‌شود.

در نهایت، تعداد انجمن‌ها را با استفاده از `max(partition.values()) + 1` به دست می‌آوریم و انجمن‌ها را به همراه آیدی اعضای آن ها چاپ می‌کنیم.

```
# Convert the graph to undirected if needed
G = G.to_undirected()

# Run the Louvain algorithm
partition = best_partition(G)

# Get the number of communities
num_communities = max(partition.values()) + 1

# Print the communities
for community_id in range(num_communities):
    nodes_in_community = [node for node, comm_id in partition.items() if comm_id
== community_id]
    print(f"Community {community_id}: {nodes_in_community}")
```

خروجی قطعه کد بالا



تعیین میزان قطبیت و حس کلی جامعه نمونه

قطبیت با استفاده از پردازش زبان طبیعیت هر تویت را بررسی کرده و بر اساس بار معنایی آن عدیی بین -۱ تا ۱ به هر تویت اختصاص میدهد. از این رو برای تعیین میزان قطبیت، قطبیت هر یک از تویت ها را با استفاده از کتابخانه textplob محاسبه و در یک لیست ذخیره کرده و در نهایت میانگین آن ها را محاسبه نمودیم.

```
# Iterate over each tweet
polarity_list = []
for tweet in df['Tweet']:
    # Calculate polarity
    blob = TextBlob(tweet)
    polarity = blob.sentiment.polarity

    # Append polarity to the list
    polarity_list.append(polarity)

# Calculate overall polarity
overall_polarity = sum(polarity_list) / len(polarity_list)

# Print overall polarity
print("Overall Polarity:", overall_polarity)
```

خروجی قطعه کد بالا

```
Overall Polarity: 0.10818496035361659
```

برای تعیین حس کلی با استفاده از داده های ذخیره شده در ستون Sentimeter_lable در دیتاست و ایجاد یک دیشکنری برای احساسات مثبت، منفی و خنثی اعداد -۱، ۰ و ۱ را لحاظ کردیم، و در نهایت برای محاسبه حس کلی از تمامی این مقادیر میانگین گرفتیم.

```
# Define the mapping dictionary
sentiment_mapping = {'negative': -1, 'neutral': 0, 'positive': 1}

# Convert 'sentiment_label' column to strings
df['sentiment_label'] = df['sentiment_label'].astype(str)

# Calculate the sentiment score for each row, handling missing values
df['sentiment_score'] = df['sentiment_label'].map(sentiment_mapping).fillna(0)

# Calculate the overall sentiment score
overall_sentiment_score = df['sentiment_score'].mean()

# Determine the overall feeling based on the sentiment score
if overall_sentiment_score < 0:
    overall_feeling = 'Negative'
elif overall_sentiment_score > 0:
    overall_feeling = 'Positive'
else:
    overall_feeling = 'Neutral'

print("Overall feeling: {}".format(overall_feeling))
```

خروجی قطعه کد بالا

تعیین تعداد خوشه ها در جامعه نمونه

برای تعیین تعداد تعداد خوشه ها در جامعه، با استفاده از قطعه کد زیر، با استفاده از الگوریتم (Louvain)، برا تشخیص اجتماعات در گراف استفاده کردیم.

```
# Use the Louvain method to detect communities
partition = community.best_partition(G)

# Count the number of unique communities
num_clusters = len(set(partition.values()))

print("Number of clusters:", num_clusters)
```

خروجی قطعه کد بالا