DSC 202 Data Management
for Data Science

Reddit Data Analysis

Group 5

Arya Rahnama
Atulya Kumar
Kaveya Sivaprakasam

Introduction

Beyond the Threads

Unveiling the Secrets of Data Science Subreddits

Discover Community Growth Strategies

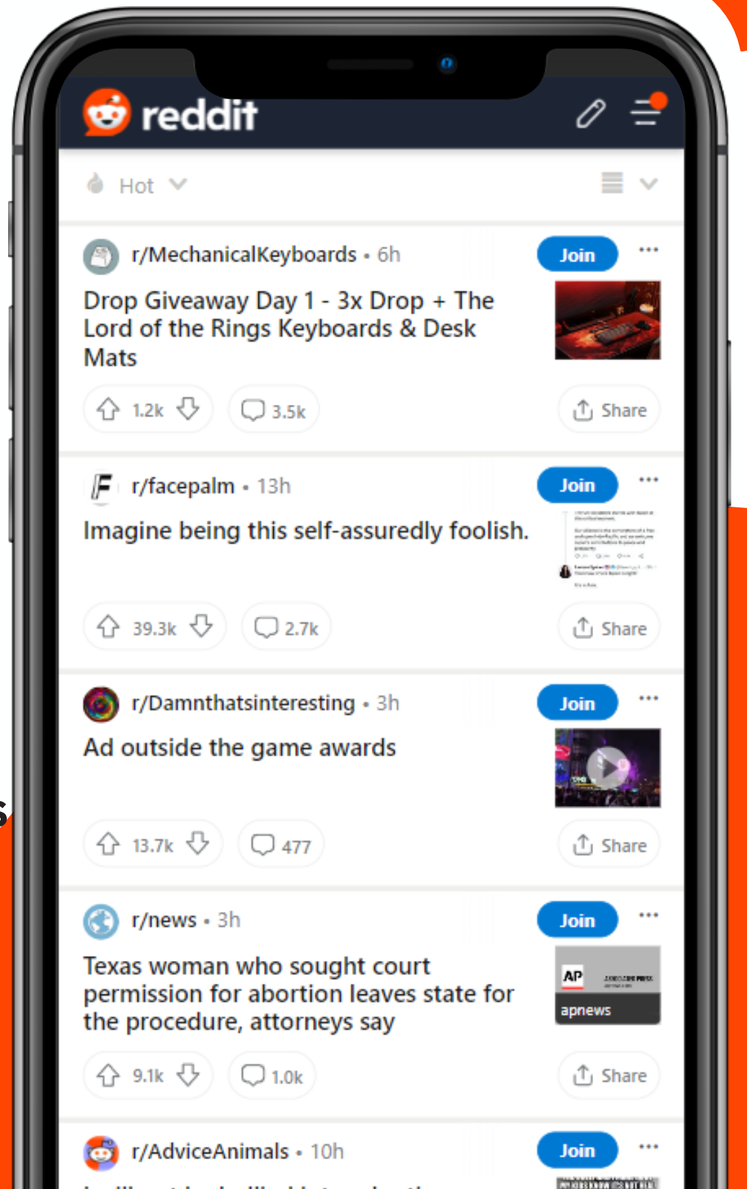
Provides a roadmap for community managers and marketers to optimize strategies for user acquisition and engagement.

Learn Content and Community Size

Guides content creators and community managers in striking the right balance for sustained growth, ensuring a vibrant and active community.

Help with Amplifying Professional Interactions

Informs individuals, businesses, and platforms about the trends and needs of professionals in various fields, fostering more meaningful connections and career-related discourse.



Data Sources

01

Relational Data

Source: kaggle

- 500,000 posts from 19 Data Science subreddits, 115MB
- The dataset could be used for various analyses, such as understanding user behavior, post popularity, or trends within specific subreddits.

02

Semi-structured Data

Source: zenodo

- Preprocessed posts from the Reddit dataset
- Each line is a JSON object representing a post.
- 3,848,330 posts, 19.6GB
- Worked on content and summary

03

Graph Data

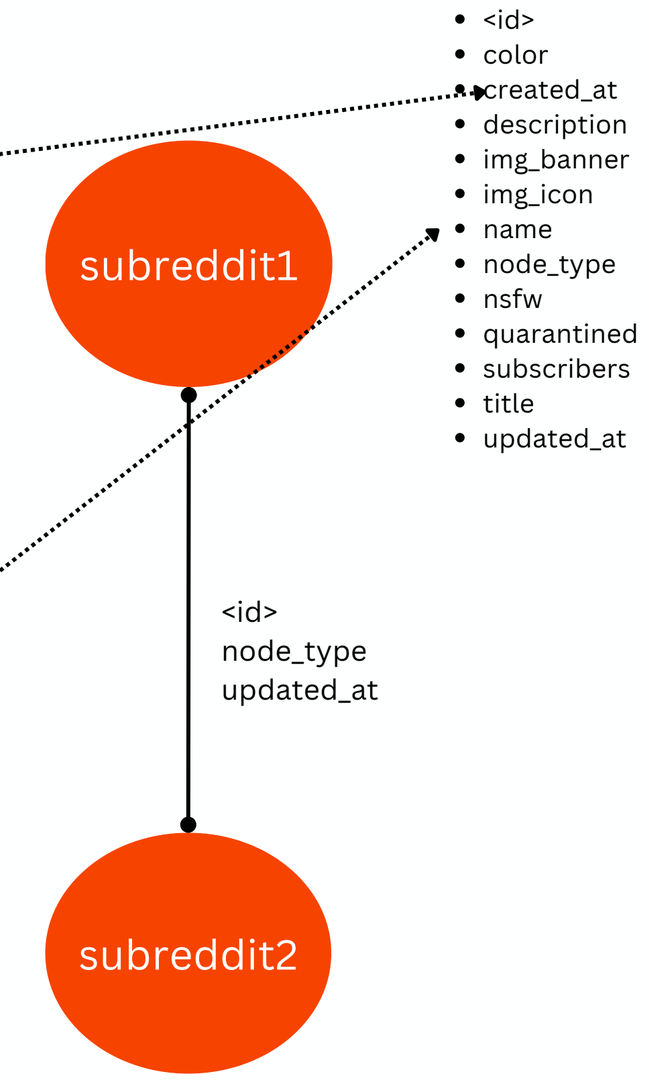
Source:kaggle

- Graph of subreddit links based on how they reference each other
- Subreddits were the nodes
- References of subreddit were the links/edges.
- 127,000 nodes and 349,744 links, 18MB

Data Model

data_science_posts		
PK	subreddit	varchar(255)
	created_date	timestamp
	created_timestamp	double precision
	title	text
	id	varchar(255)
	author	varchar(255)
	author_created_utc	double precision
	full_link	varchar(255)
	score	double precision
	num_comments	double precision
	num_crossposts	double precision
	subreddit_subscribers	double precision
	post	text

```
{
 "$schema": "http://json-schema.org/draft-04/schema#",
 "type": "object",
 "properties": {
  "_id": {
   "type": "object",
   "properties": {
    "$oid": {
     "type": "string"
    }
   },
   "required": [
    "$oid"
   ]
  },
  "author": { "type": "string" },
  "body": { "type": "string" },
  "normalizedBody": { "type": "string" },
  "content": { "type": "string" },
  "content_len": { "type": "integer" },
  "summary": { "type": "string" },
  "summary_len": { "type": "integer" },
  "id": { "type": "string" },
  "subreddit": { "type": "string" },
  "subreddit_id": { "type": "string" }
 },
 "required": ["_id", "author","body", "normalizedBody",
  "content", "content_len", "summary","summary_len",
  "id", "subreddit", "subreddit_id"
 ]
}
```



PostgreSQL Table

MongoDb Collection

Neo4j Node and Link

Sample Data

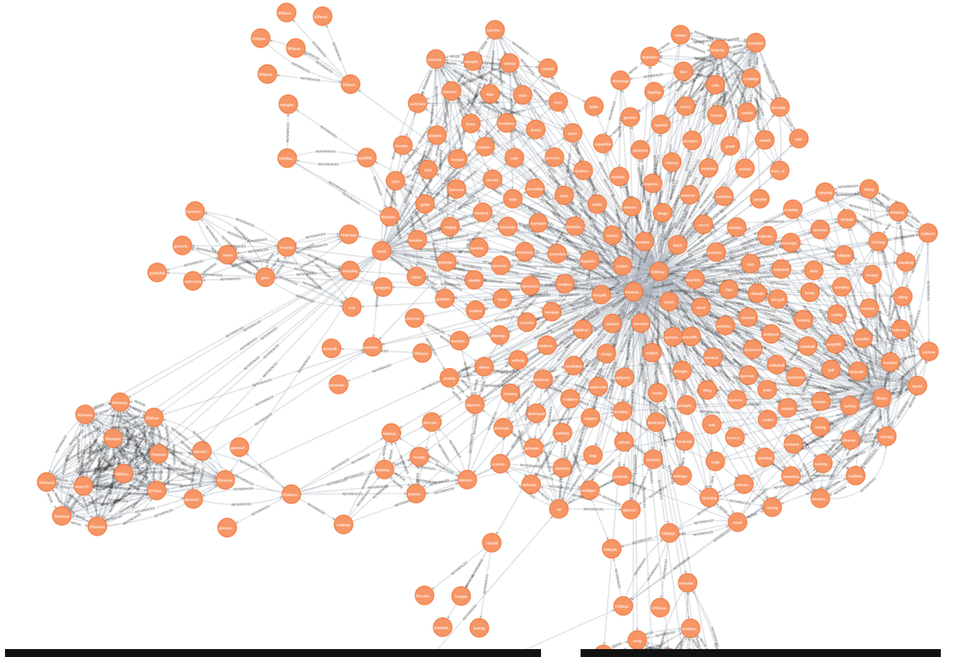
PostgreSQL Data

created_date	created_timestamp	subreddit	title	id	autho
2010-02-10 22:06:17.000000	1265832377	analytics	YouTube's traffic data for music questioned	b0ih7	salvage
2010-02-10 22:06:53.000000	1265832413	analytics	November Sees Number of U.S. Videos Viewed Online …	b0ihf	salvage
2010-02-11 19:47:22.000000	1265910442	analytics	So what do you guys all do related to analytics? W…	b0x63	xtom
2010-02-12 18:10:36.000000	1265991036	analytics	10 Web Analytics Tools For Tracking Your Visitors	b1bbg	[delete
2010-02-26 20:26:18.000000	1267208778	analytics	Improving Your Sense of Site	b6x0n	[delete
2010-03-04 20:17:26.000000	1267726646	analytics	Google's Invasive, non-Anonymized Ad Targeting: A …	b9ab7	xtom

MongoDb Data

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
6.35 GB	3.8 M	5.10 kB	1	44.58 MB


```
_id: ObjectId('656e933c4d898d74c7e68e31')
author: "raysofdarkmatter"
body: "I think it should be fixed on either UTC standard or UTC+1 year around…"
normalizedBody: "I think it should be fixed on either UTC standard or UTC+1 year around…"
content: "I think it should be fixed on either UTC standard or UTC+1 year around…"
content_len: 178
summary: "Shifting seasonal time is no longer worth it."
summary_len: 8
id: "c69al3r"
subreddit: "math"
subreddit_id: "t5_2qh0n"
```

REFERENCES

<id>	65677
node_type	sidebar
updated_at	2022-09-15 03:33:16.139093+00

Subreddit

<id>	67961
color	NaN
created_at	2009-07-29 14:34:44+00
description	NaN
img_banner	NaN
img_icon	NaN
name	machinelearning
node_type	public
nsfw	f
quarantined	f
subscribers	2536113.0
title	Machine Learning
updated_at	2022-09-14 18:49:27.98236+00

Neo4j Data

300 out of 127,801 nodes and 1349 out of 349,745 relationships visualized

Query 1

How do subreddits with the highest growth connect with other communities, and does the content summary length correlate with their growth?

```
# Query PostgreSQL for subreddits with the highest increase in subscribers
pg_cursor.execute("""
    WITH subreddit_first_last AS (
    SELECT subreddit,
           MIN(created_date) AS first_date,
           MAX(created_date) AS last_date
    FROM data_science_posts
    WHERE created_date > CURRENT_DATE - INTERVAL '3 year'
    GROUP BY subreddit
),
subreddit_first_subs AS (
    SELECT fl.subreddit,
           fl.first_date,
           fs.subreddit_subscribers AS first_subscribers
    FROM subreddit_first_last fl
    JOIN data_science_posts fs ON fl.subreddit = fs.subreddit AND fl.first_date = fs.created_date
),
subreddit_last_subs AS (
    SELECT fl.subreddit,
           fl.last_date,
           ls.subreddit_subscribers AS last_subscribers
    FROM subreddit_first_last fl
    JOIN data_science_posts ls ON fl.subreddit = ls.subreddit AND fl.last_date = ls.created_date
)
SELECT fls.subreddit,
       (lls.last_subscribers - fls.first_subscribers) AS subscriber_increase
FROM subreddit_first_subs fls
JOIN subreddit_last_subs lls ON fls.subreddit = lls.subreddit
ORDER BY subscriber_increase DESC
LIMIT 10;
""")
growing_subreddits = pg_cursor.fetchall()
```

PostgreSQL Query:
Find the top 10 data science subreddits that have experienced the highest growth in subscribers over time of 3 years.

Calculate the increase in subscribers for each subreddit.

Result:

	subreddit	subscriber_increase
1	MachineLearning	911884
2	datascience	411718
3	statistics	318231
4	computerscience	63012
5	learnmachinelearning	62642
6	dataengineering	38482
7	analytics	37338
8	artificial	31341
9	datasets	29304
10	deeplearning	19690

Further Investigation

MongoDB Query:

- Find the top 10 posts with the highest summary lengths from the subreddits identified as top growing data science subreddits in PostgreSQL


```python
# Query MongoDB for posts with the highest summary lengths from the growing subreddits
mongodb_results = collection.find(
    {"subreddit": {"$in": subreddit_names}},
    {"subreddit": 1, "summary": 1, "summary_len": 1}
).sort("summary_len", -1).limit(10)

# Extract subreddit names and summaries
subreddits_summaries = [(doc["subreddit"], doc["summary"]) for doc in mongodb_results]
```

[6]:
```python
print(subreddits_summaries)
```

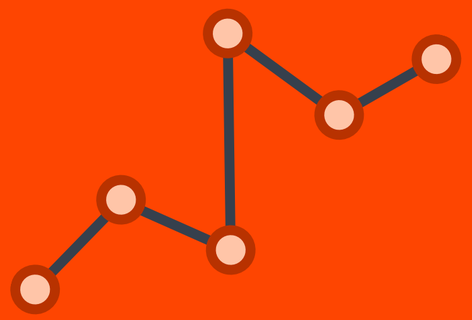
```
[('artificial', "AI becomes Enlightened. \n An Enlightened AI lends itself to several variations. The one presented al
h AI, our assumption is it is telling truth, there is no possible solution to the asteroid problem. And the upside is
some such. \n Another variation would be the Darwin AI. In this case, perhaps there is a solution, but if people can'
don't deserve to go on anyways. Terrestrial life is after all a queer but temporary result of a misguided intergalact
mewhere and went viral. \n A close variation to Darwin AI would be a Plato AI that has determined there is solution,
it will kill life on Earth, but the AI will survive, and in its judgment this is the greater Good. By unavoidable cir
Plato AI being the greatest philosopher in the land. \n What's fascinating about these scenarios is not that the AI t
it would feel to saving us. The AI could very well determine that its relationship with us is that we are imperfect,
at brought AI into existence to serve selfishly serve human desires, and this does not demand the AI's absolute loyal
```

Further Investigation

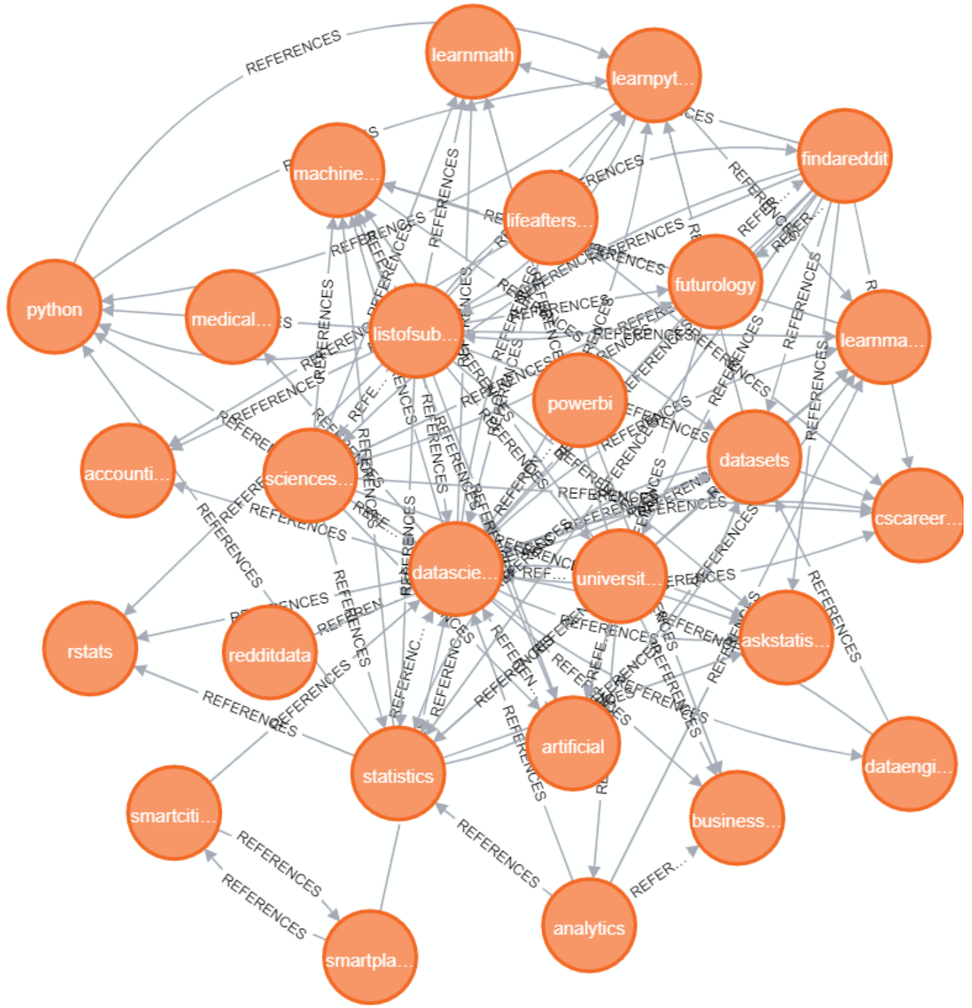
Graph Query:

- For each of the top posts, find other subreddits mentioned in their summaries and the type of relationship between them.

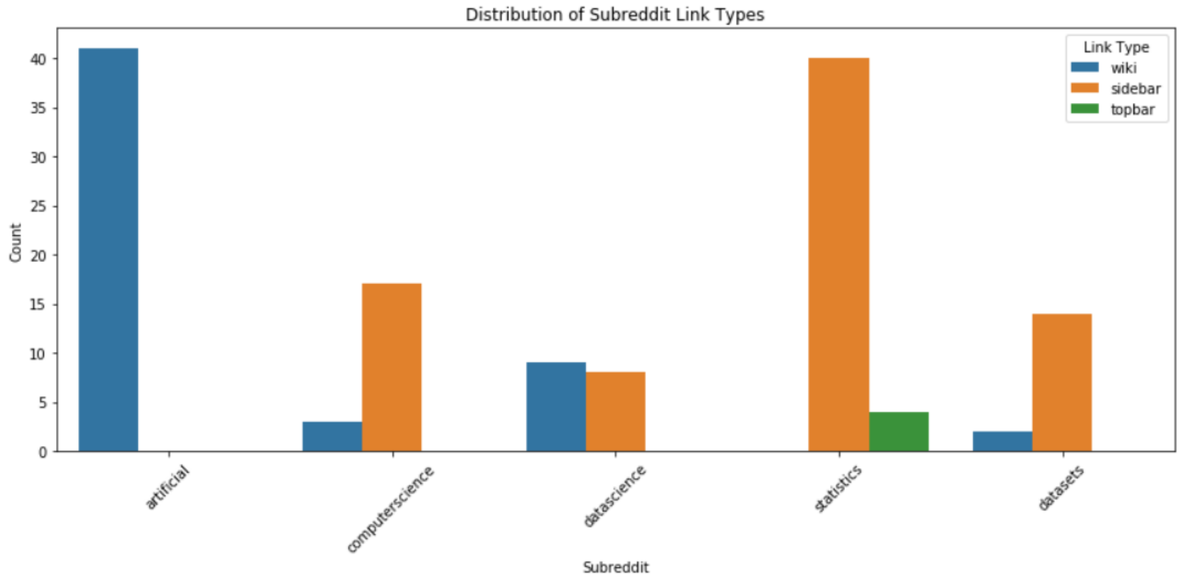
```
# Function to find other subreddits they reference and the nature of these links
def find_linked_subreddits(tx, subreddit_name):
    query = (
        "MATCH (s:Subreddit {name: $subreddit_name})-[r:REFERENCES]->(linked:Subreddit) "
        "RETURN linked.name AS linked_subreddit, r.node_type AS link_type"
    )
    result = tx.run(query, subreddit_name=subreddit_name)
    return [(record["linked_subreddit"], record["link_type"]) for record in result]

# Use the Neo4j driver to find linked subreddits for each subreddit from MongoDB results
linked_subreddits_info = {}
with neo4j_driver.session() as session:
    for subreddit, summary in subreddits_summaries:
        linked_subreddits = session.read_transaction(find_linked_subreddits, subreddit)
        linked_subreddits_info[subreddit] = linked_subreddits
```



	Subreddit	Summary	Linked Subreddits
0	artificial	AI becomes Enlightened. \n An Enlightened AI l...	[(agi, wiki), (transhumanism, wiki), (transhum...
1	computerscience	I'm a senior but I'm essentially "junior statu...	[(suggestalaptop, wiki), (learnprogramming, wi...
2	datascience	My concern is that I don't think I'm going to...	[(python, wiki), (medical_datascience, wiki), ...
3	statistics	Need help with estimating population with a sa...	[(datasets, topbar), (rstats, sidebar), (pytho...
4	datasets	Started a club in HS for computer programming ...	[(sportsreference, wiki), (bigquery, wiki), (w...

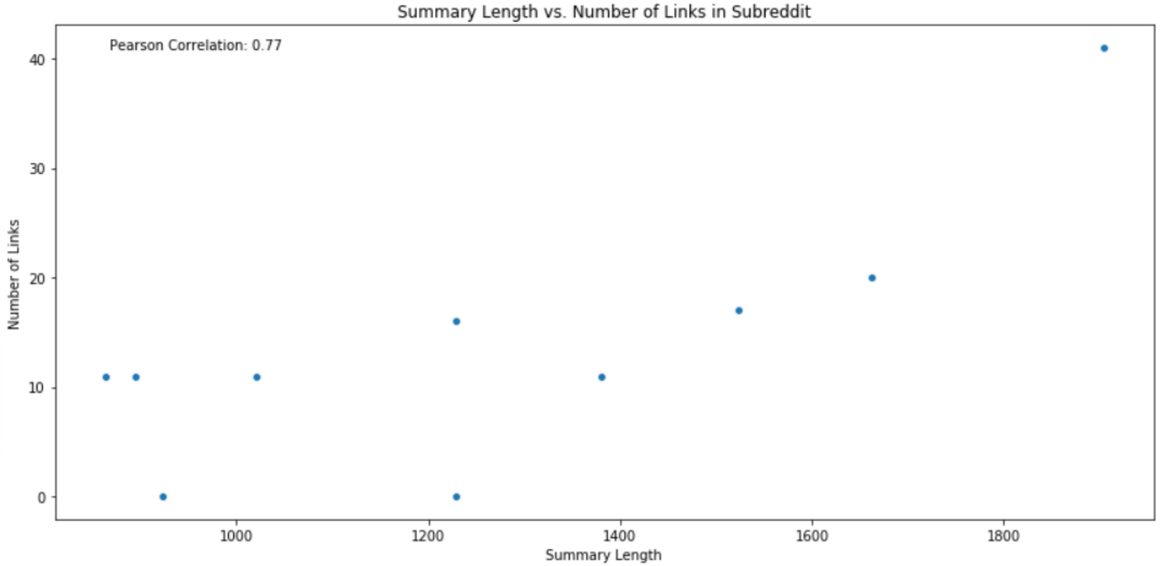


What we found?



Distribution of Subreddit Link Types:

- This bar chart displays the count of different types of links (wiki, sidebar, topbar) associated with each subreddit.
- It appears that certain link types are more prevalent in some subreddits compared to others.



Summary Length vs. Number of Links in Subreddit:

- The strong positive correlation suggests that as subreddits grow and have longer summary content, they also tend to be more interconnected with other subreddits through various types of links.

Query 2

Is there a relationship between the volume of content and the network size of rapidly growing subreddits?

Postgres Query
Find the subreddits that have experienced the most significant growth in posting frequency over the last three years.

```
# Query to find subreddits with the largest increase in posting frequency over the last 3 years
pg_cursor.execute("""
    SELECT subreddit, COUNT(*) AS post_count
    FROM data_science_posts
    WHERE created_date BETWEEN CURRENT_DATE - INTERVAL '3 year' AND CURRENT_DATE
    GROUP BY subreddit
    ORDER BY post_count DESC
    LIMIT 10;
""")
subreddit_growth = pg_cursor.fetchall()
```

	subreddit	post_count
1	MachineLearning	32221
2	datascience	21882
3	statistics	18805
4	computerscience	16914
5	learnmachinelearning	16118
6	AskStatistics	11150
7	dataengineering	8415
8	artificial	7459
9	deeplearning	6089
10	DataScienceJobs	5941

Query 2

Is there a relationship between the volume of content and the network size of rapidly growing subreddits?

MongoDB Query

Matches posts from subreddits identified in the previous PostgreSQL query.
Groups the data by subreddit, calculating the average content length and counting the total number of posts for each subreddit.

```
# Query to count the number of posts and calculate average content lengt
subreddit_content_data = collection.aggregate([
    {"$match": {"subreddit": {"$in": subreddit_names_growth}}},
    {"$group": {
        "_id": "$subreddit",
        "average_content_len": {"$avg": "$content_len"},
        "post_count": {"$sum": 1}
    }}
])
```

	_id	average_content_len	post_count
0	computerscience	218.421687	83
1	statistics	221.500000	344
2	artificial	258.431373	51
3	MachineLearning	210.335294	170
4	datascience	249.800000	75
5	AskStatistics	224.775862	58
6	DataScienceJobs	68.000000	1

Query 2

Is there a relationship between the volume of content and the network size of rapidly growing subreddits?

Graph Query

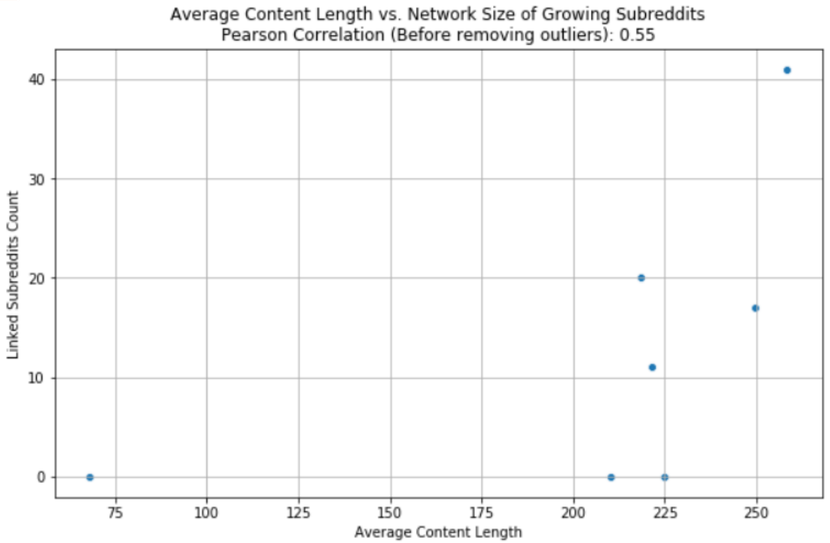
Query Neo4j graph database to find the number of linked subreddits for each subreddit that was identified as having the largest increase in posting frequency in the initial PostgreSQL query. It then adds this information to the DataFrame obtained from the MongoDB query

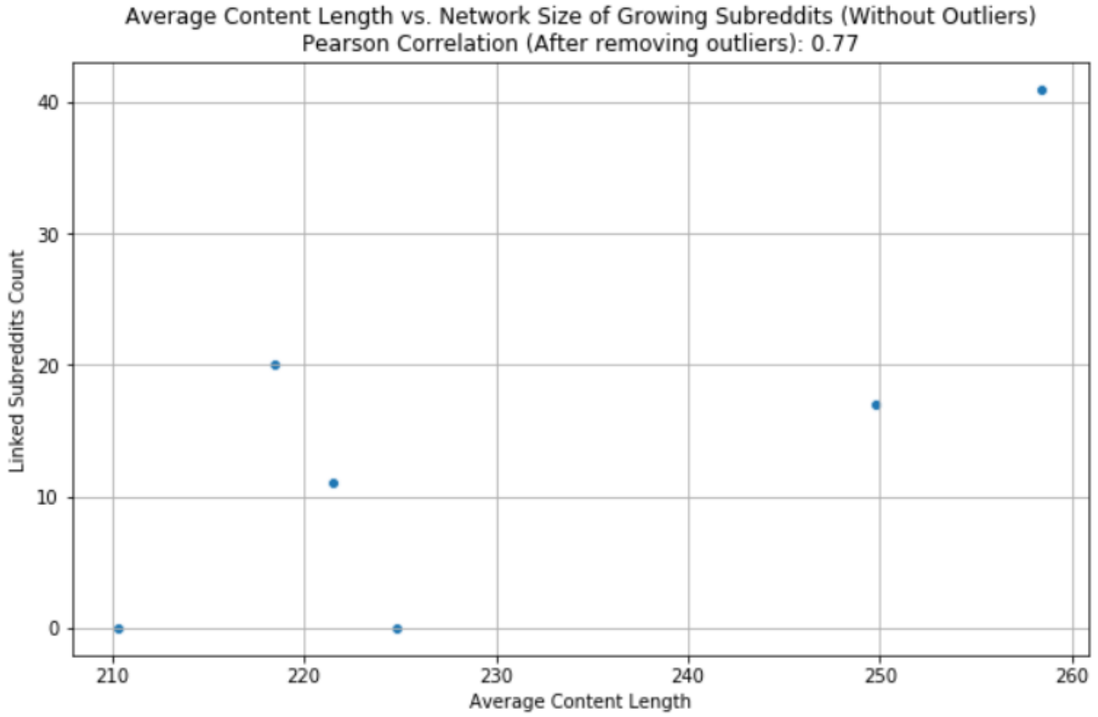
```python
# Function to find the number of linked subreddits for each growing subreddit
def find_linked_subreddits_count(tx, subreddit_name):
    query = (
        "MATCH (s:Subreddit {name: $subreddit_name})-[:REFERENCES]->(linked:Subreddit) "
        "RETURN COUNT(linked) AS linked_subreddit_count"
    )
    result = tx.run(query, subreddit_name=subreddit_name)
    return result.single()[0]

# Query Neo4j for each subreddit's linked subreddits count
subreddit_links_counts = {}
with neo4j_driver.session() as session:
    for subreddit in subreddit_names_growth:
        linked_count = session.read_transaction(find_linked_subreddits_count, subreddit)
        subreddit_links_counts[subreddit] = linked_count
```

	_id	average_content_len	post_count	linked_subreddits_count
0	computerscience	218.421687	83	20
1	statistics	221.500000	344	11
2	artificial	258.431373	51	41
3	MachineLearning	210.335294	170	0
4	datascience	249.800000	75	17
5	AskStatistics	224.775862	58	0
6	DataScienceJobs	68.000000	1	0

What we found?

- Initially, we observed a Pearson correlation coefficient of 0.55.
- However, after removing outliers in content length, the correlation coefficient increased to 0.77.
- This higher correlation coefficient after filtering outliers implies a stronger positive relationship, indicating that subreddits with longer posts tend to have more connections to other subreddits.





Query 3

How have the oldest data science subreddits influenced the **evolution** of newer subreddits in the same field?

PostgresQuery
Identify and retrieve information about the oldest data science subreddits based on the date of their first posts.

```
# Query to identify the oldest data science subreddits
pg_cursor.execute("""
    SELECT subreddit, MIN(created_date) AS first_post_date
    FROM data_science_posts
    GROUP BY subreddit
    ORDER BY first_post_date
    LIMIT 5;
""")
```

	subreddit	first_post_date
1	statistics	2008-03-19 10:08:43.000000
2	computerscience	2008-06-23 18:50:07.000000
3	artificial	2008-07-06 16:00:14.000000
4	MachineLearning	2009-07-29 17:35:16.000000
5	rstats	2009-10-02 06:01:47.000000

Query 3

How have the oldest data science subreddits influenced the **evolution** of newer subreddits in the same field?

Query a Neo4j graph database to find newer data science subreddits that are influenced by the top 5 oldest subreddits identified in the previous PostgreSQL query.

```python
# Function to find newer data science subreddits influenced by the oldest ones
def get_influenced_subreddits(tx, old_subreddits):
    query = """
    MATCH (old:Subreddit)-[:REFERENCES]->(new:Subreddit)
    WHERE old.name IN $old_subreddits
    RETURN new.name AS subreddit, COUNT(*) AS influence_count
    ORDER BY influence_count DESC
    """
    result = tx.run(query, old_subreddits=old_subreddits)
    return [(record["subreddit"], record["influence_count"]) for record in result]

# Query Neo4j and store the results
with neo4j_driver.session() as session:
    influenced_subreddits = session.read_transaction(get_influenced_subreddits, oldest_subreddit_names)
```

['compsci', 'machinelearning', 'datascience', 'learnprogramming', 'datasets', 'agi', 'transhumanism', 'transhuman', 'singularity', 'simulate', 'robotics', 'opencog', 'neurophilosophy', 'neuralnetworks', 'mlquestions', 'mlclass', 'learnmachinelearning', 'languagetechnology', 'healthai', 'genetic_algorithms', 'gameai', 'friendlyai', 'evolutionarycomp', 'evocomp', 'datascienceprojects', 'datasciencenews', 'datasciencejobs', 'datamining', 'controltheory', 'controlproblem', 'computervision', 'compressivesensing', 'cogsci', 'automate', 'art_int', 'artificialinteligence', 'alife', 'aivsai', 'aivideos', 'aiml', 'aihub', 'aiethics', 'aiclass', 'suggestalaptop', 'buildapc', 'theoreticalcs', 'techsupport', 'technology', 'programminglanguages', 'programming', 'opensource', 'linuxquestions', 'linux', 'electronics', 'ece', 'csmajors', 'cscareerquestions', 'computerengineering', 'codinghelp', 'askcomputerscience', 'rstats', 'python', 'dataisbeautiful', 'computerscience', 'biostatistics', 'askstatistics']

Query 3

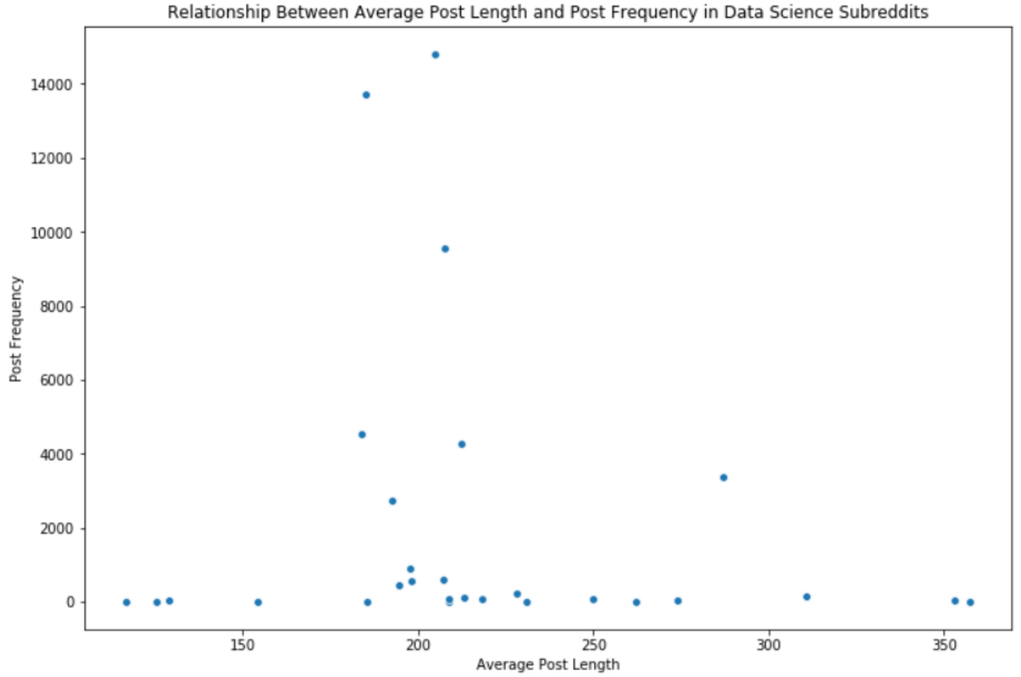
How have the oldest data science subreddits influenced the **evolution** of newer subreddits in the same field?

Query a MongoDB collection to calculate the average post length and post frequency for the newer data science subreddits that are influenced by the top 5 oldest subreddits

```
# Query to calculate the average post length and frequency for the influenced
mongodb_query = [
    {"$match": {"subreddit": {"$in": influenced_subreddit_names}}},
    {"$group": {
        "_id": "$subreddit",
        "average_post_length": {"$avg": "$content_len"},
        "post_frequency": {"$sum": 1}
    }}
]

influenced_subreddits_data = list(collection.aggregate(mongodb_query))
```

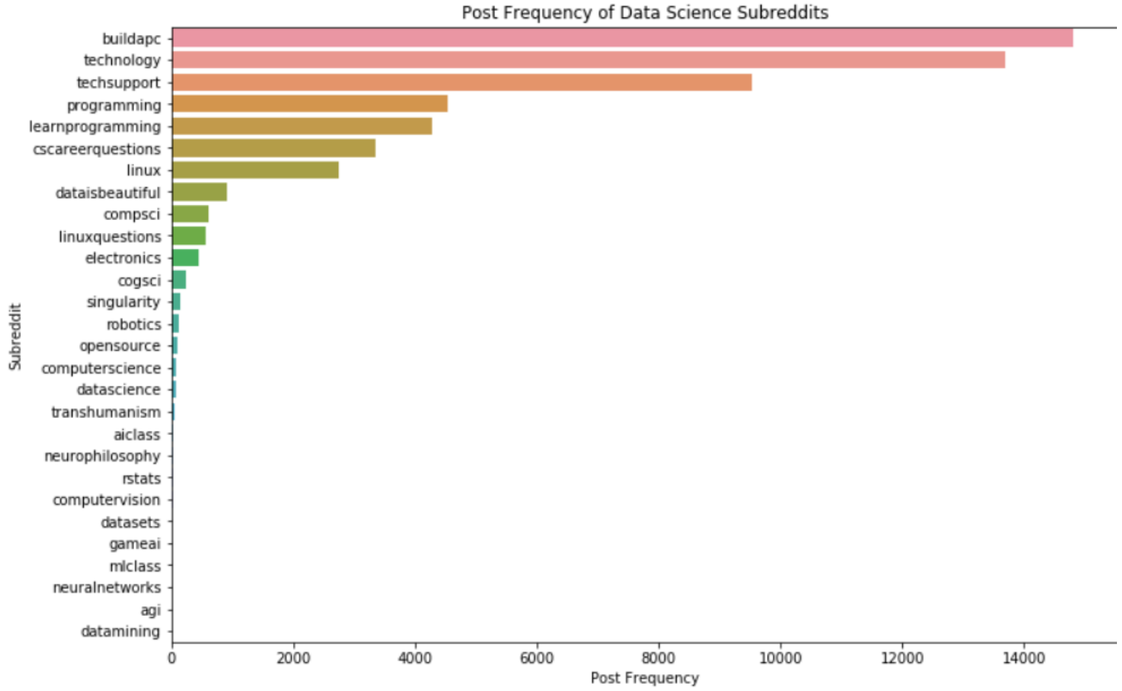
	_id	average_post_length	post_frequency
0	aiclass	128.758621	29
1	transhumanism	274.078431	51
2	mlclass	116.750000	4
3	linuxquestions	198.010508	571
4	datasets	154.230769	13
5	neuralnetworks	357.333333	3
6	singularity	310.735714	140
7	cogsci	228.191304	230
8	dataisbeautiful	197.641758	910
9	datamining	262.000000	1
10	robotics	213.169355	124

What we found?

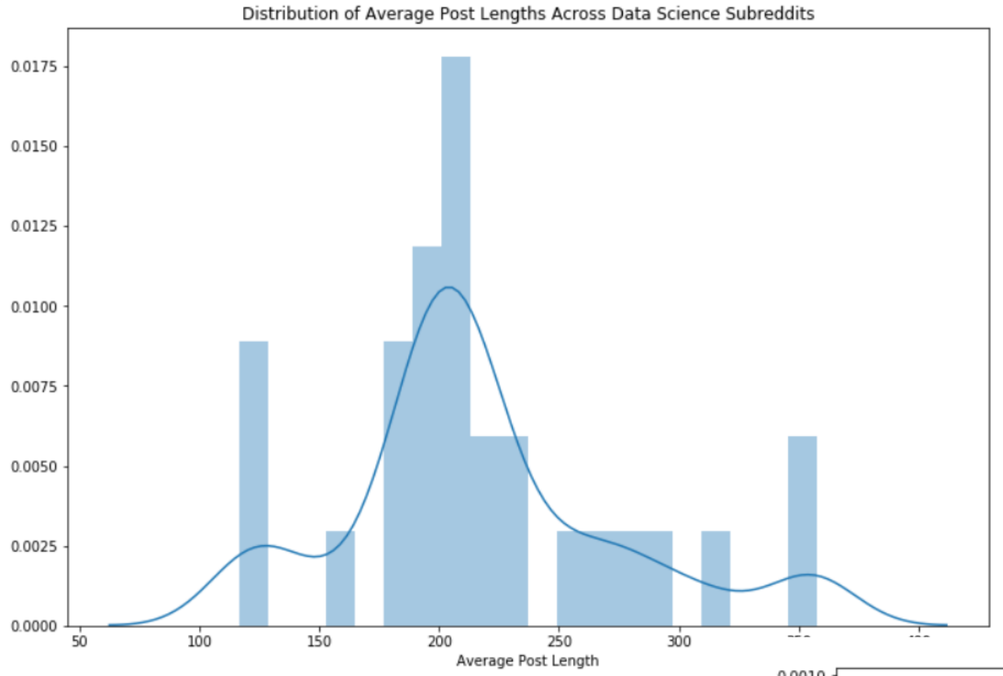


Subreddits with medium average post lengths tend to have higher post frequencies. This could indicate that communities with more in-depth discussions are also more active, and that active communities tend to normalize their post length on average.

The bar chart ranks subreddits by post frequency and highlights the most active communities. there are vibrant hubs for data science-related discussions.

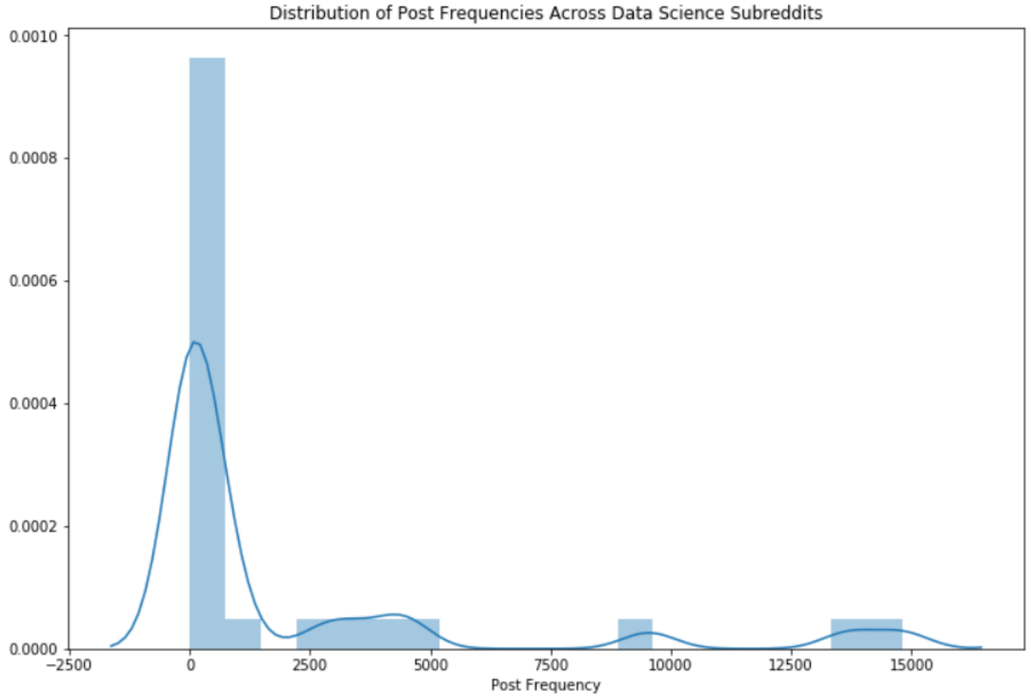


What we found?



The distribution of average post lengths shows a concentration around lower word counts, with fewer subreddits featuring longer posts on average.

Meanwhile, the post frequency distribution is heavily skewed, with a few subreddits exhibiting exceptionally high activity levels, which may point to them being key influencers within the data science community on Reddit.



Query 4

What is the relationship between career discussion frequency, subreddit centrality, and the depth of **career-related discourse?**

Postgres Query

Identify and display posts related to careers within the field of data science. The PostgreSQL query searches for posts whose titles contain keywords associated with career-related topics such as job opportunities, interviews, resumes, and hiring.

```
# Query to identify career-related posts
pg_cursor.execute("""
SELECT id, subreddit, title, created_date
FROM data_science_posts
WHERE title ILIKE ANY (array['%career%', '%job%', '%interview%', '%resume%', '%hiring%'])
ORDER BY created_date DESC;
""")
```

	ID	Subreddit	Title	CreatedDate
0	ul2fww	datascience	I just currently finishing my MSc and am about...	2022-05-08 17:05:17
1	ul0kyp	datascience	Skill set for Data science intern Interview	2022-05-08 15:23:53
2	ukzhkp	datascience	How likely can I land a job in Europe as a mid...	2022-05-08 14:10:51
3	ukwzvo	dataengineering	I have a DE interview in a few days and lookin...	2022-05-08 11:00:26
4	ukvhal	computerscience	Starting school and a career in computer scien...	2022-05-08 09:07:41

Query 4

What is the relationship between career discussion frequency, subreddit centrality, and the depth of **career-related discourse?**

Graph Query

Examine the links between career-focused subreddits and data subreddits in a Neo4j graph database.

Finds connections where a career-focused subreddit references an data subreddit containing the term 'data'.

```python
# Function to examine links between career-focused and educational subreddits
def get_career_subreddit_links(tx, career_subreddits):
    query = """
    MATCH (s:Subreddit)-[:REFERENCES]->(target:Subreddit)
    WHERE s.name IN $career_subreddits AND target.name CONTAINS 'data'
    RETURN s.name AS source, target.name AS target, COUNT(*) AS link_count
    ORDER BY link_count DESC;
    """
    result = tx.run(query, career_subreddits=career_subreddits)
    return [(record["source"], record["target"], record["link_count"]) for record in result]

# Assuming career_subreddits is a list of subreddit names obtained from the PostgreSQL query
career_subreddits = career_posts_df['Subreddit'].tolist()  # Modify this line to get actual subreddit names
with neo4j_driver.session() as session:
    career_links_data = session.read_transaction(get_career_subreddit_links, career_subreddits)
```

	SourceSubreddit	TargetSubreddit	LinkCount
0	statistics	datasets	2
1	datascience	medical_datascience	1
2	datasets	opendata	1
3	dataengineering	database	1
4	datascience	dataengineering	1
5	datasets	datahoarder	1
6	statistics	dataisbeautiful	1

Query 4

What is the relationship between career discussion frequency, subreddit centrality, and the depth of **career-related discourse?**

Sentiment Analysis

Perform sentiment analysis on the text content of posts from career-related subreddits in a MongoDB collection.

The code uses the TextBlob library to perform sentiment analysis on the content of each post.

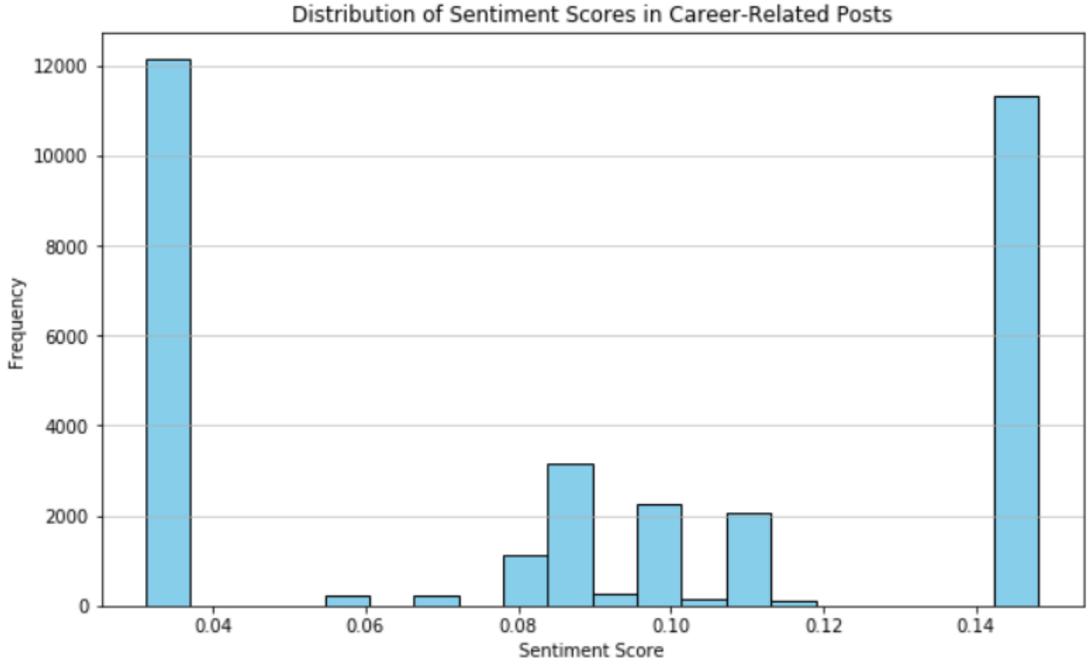
Sentiment is calculated using the polarity score, which represents the positivity or negativity of the text. For each subreddit, the average sentiment score is calculated.

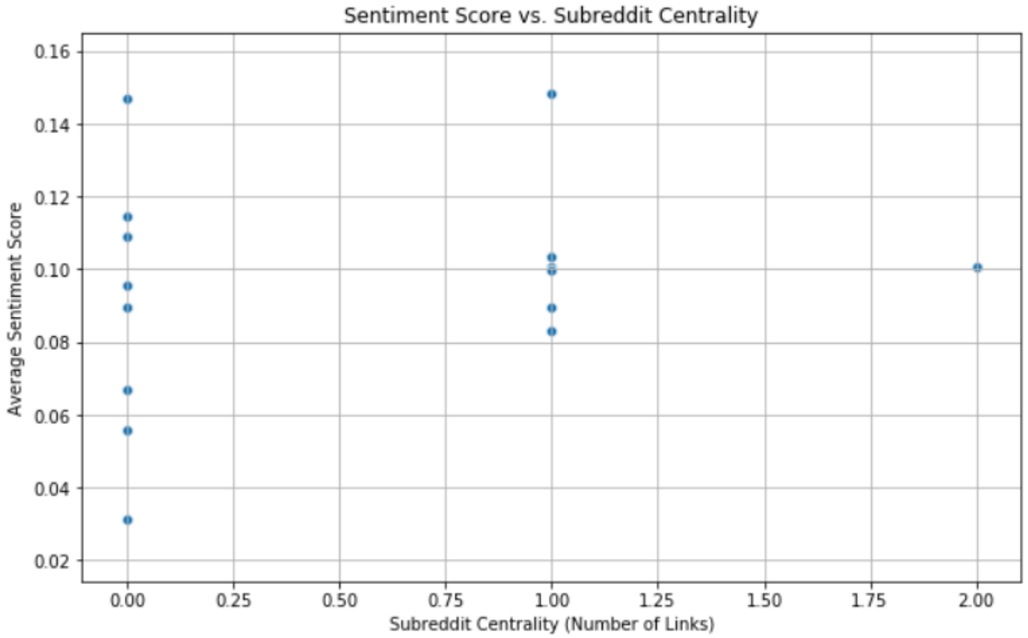
```
# Retrieve the text content of each post for sentiment analysis
posts_for_sentiment = collection.find(
    {"subreddit": {"$in": career_subreddits}},
    {"content": 1, "subreddit": 1}
)

# Initialize a dictionary to store the sentiment analysis results
sentiment_results = {}

# Perform sentiment analysis on each post's content
for post in posts_for_sentiment:
    # Assume 'content' field contains the text of the posts
    analysis = TextBlob(post["content"]) if 'content' in post else None
    if analysis:
        # For simplicity, consider polarity as sentiment
        sentiment = analysis.sentiment.polarity
        subreddit = post["subreddit"]
        if subreddit in sentiment_results:
            sentiment_results[subreddit].append(sentiment)
        else:
            sentiment_results[subreddit] = [sentiment]

# Calculate the average sentiment for each subreddit
for subreddit, sentiments in sentiment_results.items():
    sentiment_results[subreddit] = sum(sentiments) / len(sentiments) if sentiments else 0
```

	ID	Subreddit	Title	CreatedDate	average_sentiment
0	ul2fww	datascience	I just currently finishing my MSc and am about...	2022-05-08 17:05:17	0.148180
1	ul0kyp	datascience	Skill set for Data science intern Interview	2022-05-08 15:23:53	0.148180
2	ukzhkp	datascience	How likely can I land a job in Europe as a mid...	2022-05-08 14:10:51	0.148180
3	ukwzvo	dataengineering	I have a DE interview in a few days and lookin...	2022-05-08 11:00:26	NaN
4	ukvhal	computerscience	Starting school and a career in computer scien...	2022-05-08 09:07:41	0.147168

What we found?

histogram of sentiment scores in career-related posts suggests a bimodal distribution, with a **concentration** of sentiment scores towards **both ends**



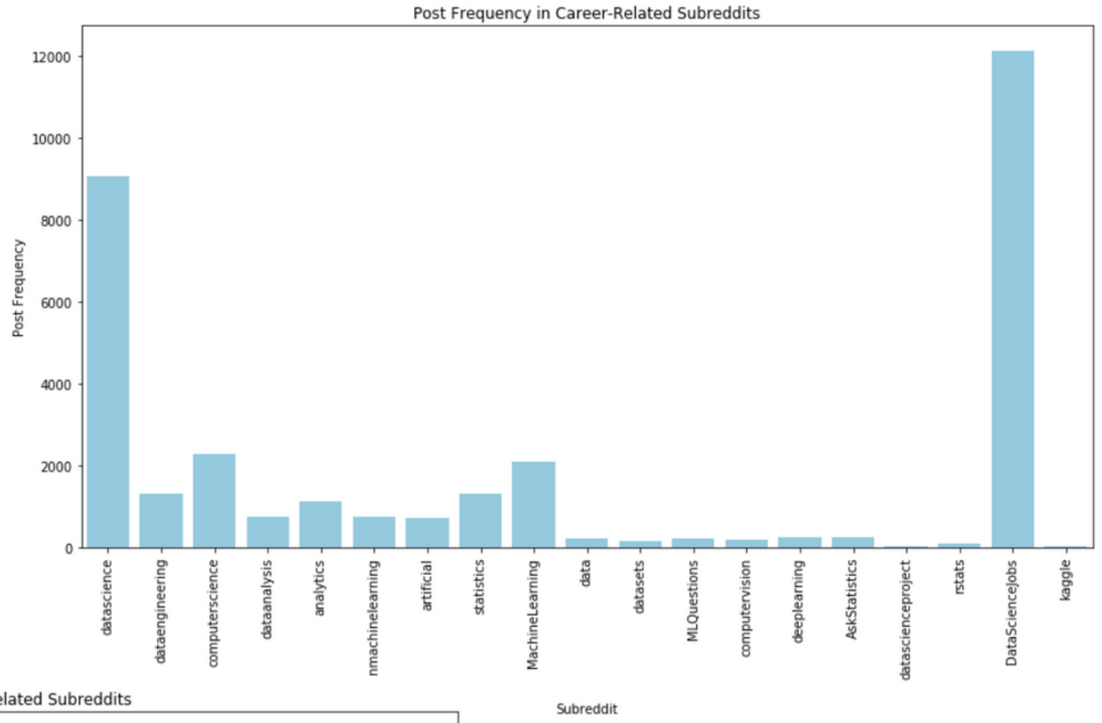


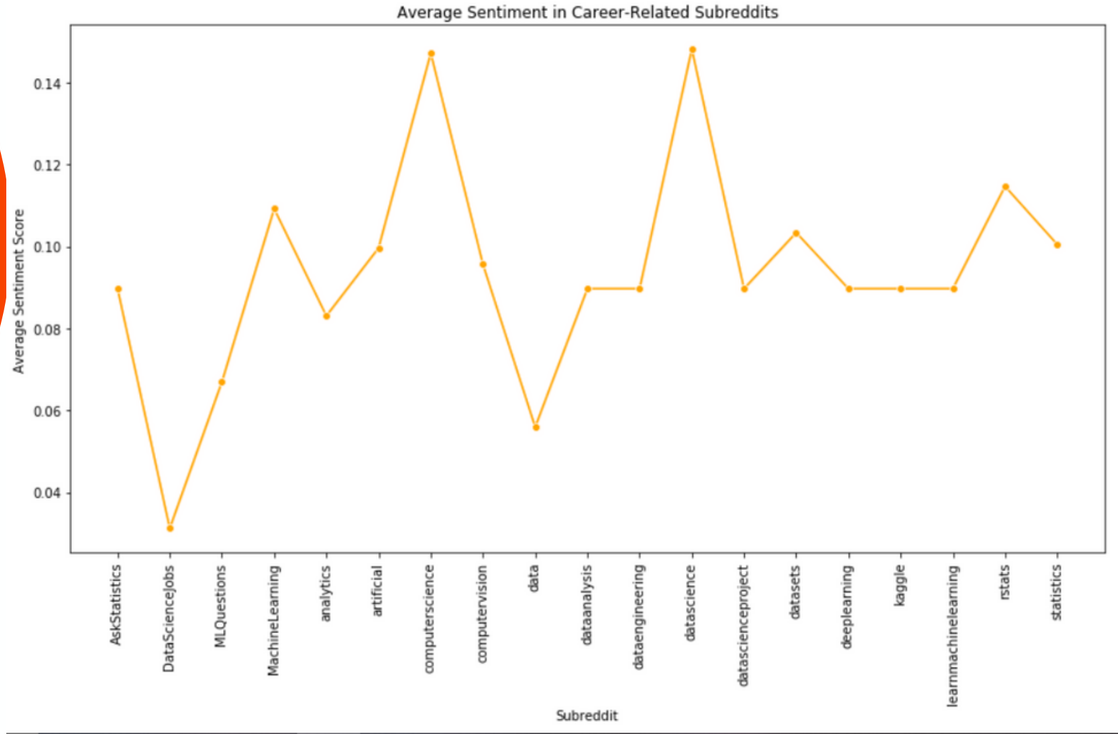
scatter plot comparing subreddit centrality to average sentiment scores **does not** demonstrate a direct relationship.

What we found?

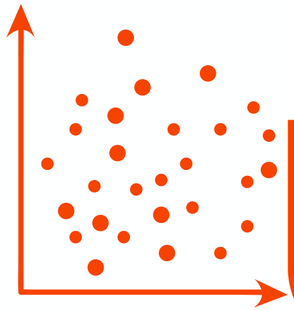
This highlights the complexity of factors that contribute to the emotional tone of discussions, beyond just the quantity of conversation.

certain subreddits are hotspots for career-related discussions





frequency of posts does not appear to be correlated with sentiment



Happy Holidays!

REFERENCES AND TOOLS USED

- **MongoDb/json data:** https://zenodo.org/records/1043504#.Wzt7PbhXryo
- **PostgreSQL/relational data:** https://www.kaggle.com/datasets/maksymshkliarevskyi/reddit-data-science-posts
- **Neo4j/graph data**: https://www.kaggle.com/datasets/thedevastator/all-subreddits-and-relations-between-them/
- **TextBlob Sentiment Analysis Documentation:** https://textblob.readthedocs.io/en/dev/
- **Docker setup neo4j:** https://hub.docker.com/_/neo4j
- **Docker setup postgres**: https://hub.docker.com/_/postgres
- **Docker setup mongoDb:** https://hub.docker.com/_/mongo