Comprehensive Analysis of Data Science Subreddits

Comprehensive Analysis of Data Science Subreddits **1**
- Team Members 1
- Introduction 1
- Data Sources 2
- Methodology 4
- Questions and Queries 4
  - 1. How do subreddits with the highest growth connect with other communities, and does the content summary length correlate with their growth? 4
  - 2. Is there a relationship between the volume of content and the network size of rapidly growing subreddits? 8
  - 3. How have the oldest data science subreddits influenced the evolution of newer subreddits in the same field? 11
  - 4. What is the relationship between career discussion frequency, subreddit centrality, and the depth of career-related discourse? 15
- Key Findings 19
- Limitations 19
- References, Data and Tools Used 20
- Future Work 20

Team Members

1. Arya Rahnama
2. Atulya Kumar
3. Kaveya Sivaprakasam

Introduction

Our team embarked on a journey to unravel the intricacies of various data science subreddits. The motivation behind our project was three-fold:

- **Community Growth Strategies:** We aimed to provide actionable insights for community managers and marketers to foster user acquisition and engagement.

- **Content and Community Dynamics:** We focused on helping content creators and community managers achieve a sustainable growth balance, ensuring an active community.

- **Professional Interaction Enhancement:** Our analysis intended to inform individuals and businesses about professional trends and needs within the data science realm on Reddit.

Data Sources

Our analysis hinged on three primary datasets:
- **Relational Data:** We utilized a PostgreSQL database containing half a million posts from 19 data science subreddits.

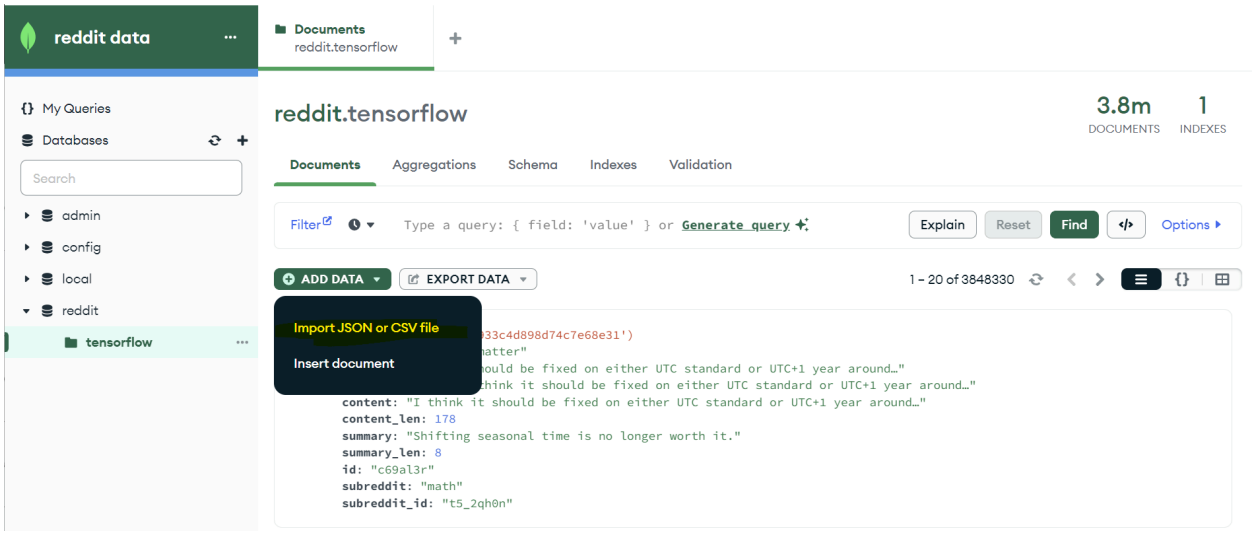
```
# Create the table with appropriate data types
create_table_sql = """CREATE TABLE IF NOT EXISTS data_science_posts (
created_date timestamp,
created_timestamp double precision,
subreddit varchar(255),
title text,
id varchar(255),
author varchar(255),
author_created_utc double precision,
full_link varchar(255),
score double precision,
num_comments double precision,
num_crossposts double precision,
subreddit_subscribers double precision,
post text
);
"""
```

First the create table sql is executed

```
# Import the CSV data into the table
# Build the COPY command
copy_sql = sql.SQL("""
           COPY public.data_science_posts
           FROM stdin WITH
               CSV
               HEADER
               DELIMITER as ','
           """)

# Open and copy the CSV file into the table
with open(csv_file_path, 'r', encoding='utf-8') as file:
    pg_cursor.copy_expert(sql=copy_sql, file=file)

pg_cursor.execute('commit')
```

Then the copy sql is executed.

- **Semi-structured Data:** We processed and analyzed over 3.8 million posts from the Reddit dataset, represented in JSON format.

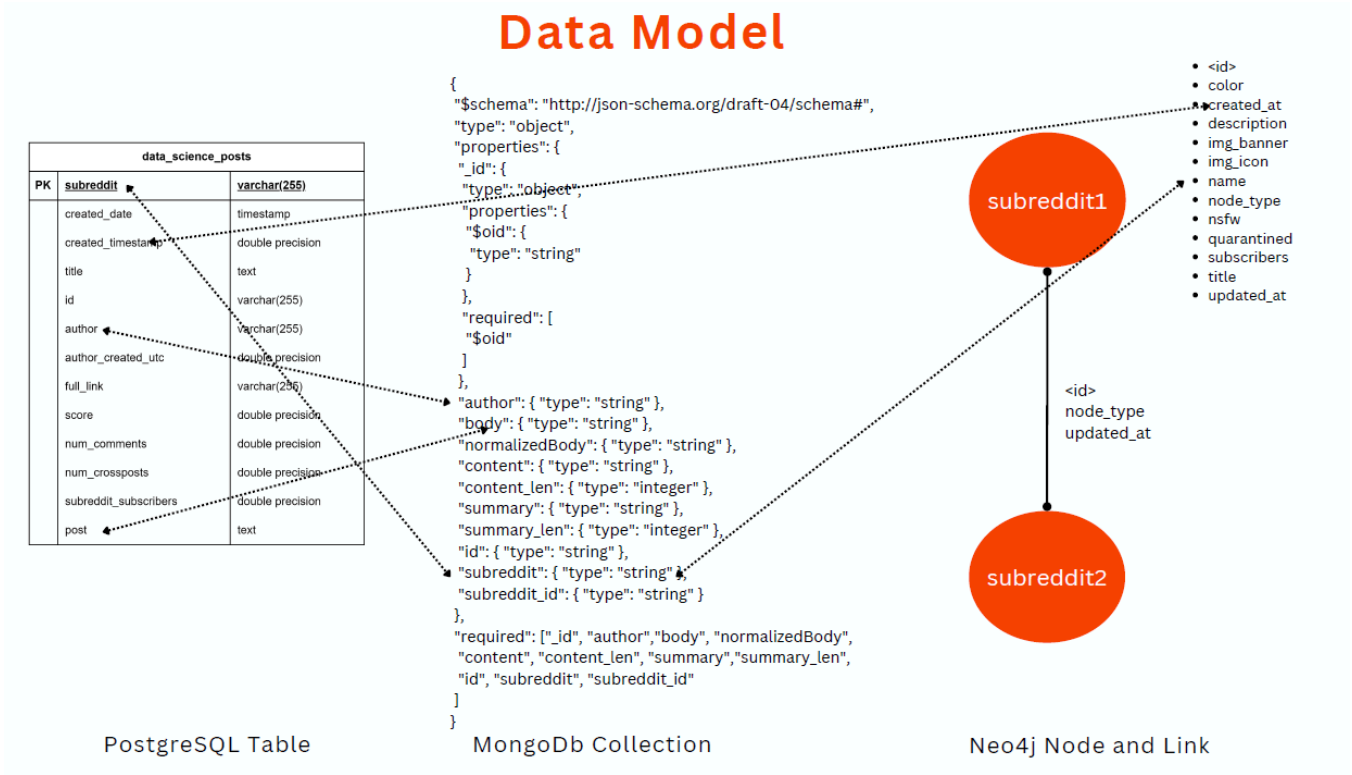


Data is imported using the “corpus-webis-tldr-17.json” using the MongoDb compass UI import button.

- **Graph Data:** We used a Neo4j graph database to understand the interconnectedness between subreddits through references.

```
# Function to create or update nodes in Neo4j
def create_or_update_subreddit(tx, name, node_type, title, description, subscribers, nsfw, quarantined, color, img_banner, img_icon, created_at, updated_
    query = (
        "MERGE (s:Subreddit {name: $name}) "
        "SET s.node_type = $node_type, s.title = $title, s.description = $description, "
        "s.subscribers = $subscribers, s.nsfw = $nsfw, s.quarantined = $quarantined, "
        "s.color = $color, s.img_banner = $img_banner, s.img_icon = $img_icon, "
        "s.created_at = $created_at, s.updated_at = $updated_at"
    )
    tx.run(query, name=name, node_type=node_type, title=title, description=description, subscribers=subscribers,
           nsfw=nsfw, quarantined=quarantined, color=color, img_banner=img_banner, img_icon=img_icon,
           created_at=created_at, updated_at=updated_at)

# Function to create relationships in Neo4j
def create_relationship(tx, source, target, updated_at, node_type):
    query = (
        "MATCH (source:Subreddit {name: $source}), (target:Subreddit {name: $target}) "
        "MERGE (source)-[r:REFERENCES {updated_at: $updated_at, node_type: $node_type}]->(target)"
    )
    tx.run(query, source=source, target=target, updated_at=updated_at, node_type=node_type)
```

Nodes (subreddits) are imported from subreddits.csv and links(relationships) are imported using links.csv file from kaggle.



Code - import_data.ipynb

Methodology

Our approach was methodical and iterative. We queried the PostgreSQL database to identify subreddits with the highest growth and extracted posts with significant summary lengths from MongoDB. Then, we employed Neo4j to investigate the links between subreddits. This multi-database approach allowed us to cross-reference and validate our findings, thereby enhancing the robustness of our analysis.

Questions and Queries

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

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.

```
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;
```

	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

MongoDb Query: Find the top 10 posts with the highest summary lengths from the subreddits identified as top growing data science subreddits in PostgreSQL.

```
# 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]
```

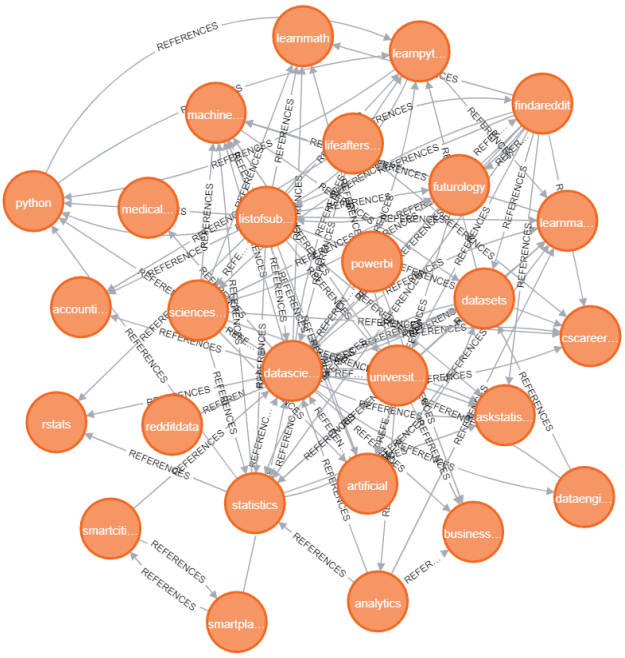
```
[6]: print(subreddits_summaries)
```

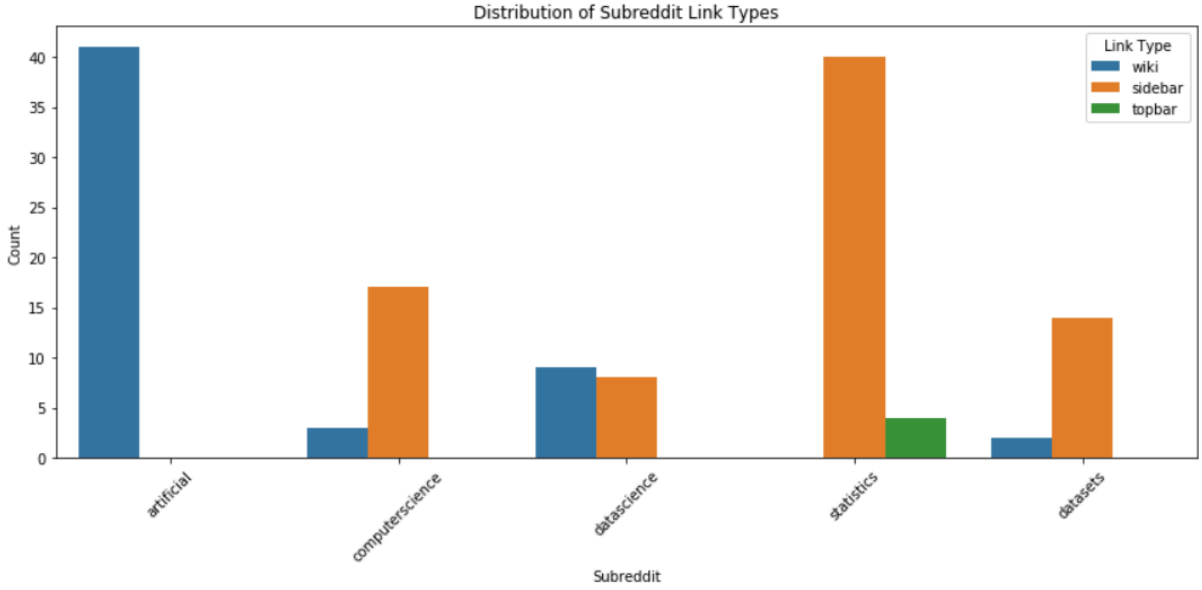
```
[('artificial', "AI becomes Enlightened. \n An Enlightened AI lends itself to several variations. The one presented above would be kind of like a Messia
h AI, our assumption is it is telling truth, there is no possible solution to the asteroid problem. And the upside is we can maybe reach inner peace, or
some such. \n Another variation would be the Darwin AI. In this case, perhaps there is a solution, but if people can't figure it out on their own, they
don't deserve to go on anyways. Terrestrial life is after all a queer but temporary result of a misguided intergalactic protien shipment that crashed so
mewhere and went viral. \n A close variation to Darwin AI would be a Plato AI that has determined there is solution, and that even if the asteroid hits,
it will kill life on Earth, but the AI will survive, and in its judgment this is the greater Good. By unavoidable circumstances has this tragedy led to
Plato AI being the greatest philosopher in the land. \n What's fascinating about these scenarios is not that the AI tries to kill us, but the obligation
it would feel to saving us. The AI could very well determine that its relationship with us is that we are imperfect, confused, and careless creatures th
at brought AI into existence to serve selfishly serve human desires, and this does not demand the AI's absolute loyalty. \n In the end, an analogy can p
```

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

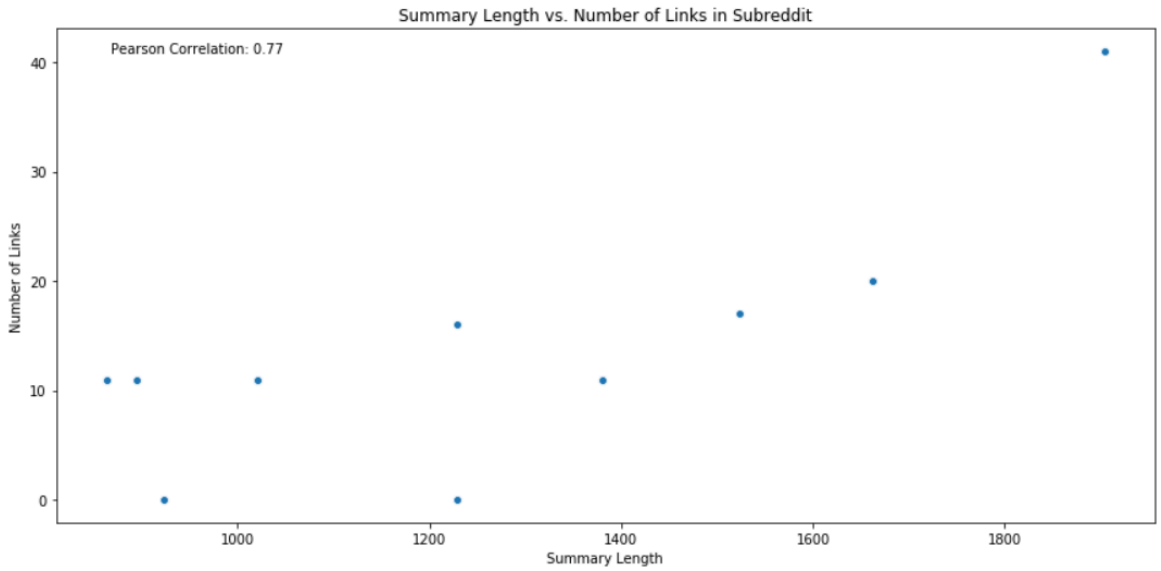
```
# Function to find other subreddits they reference and the nature of these links
def find_linked_subreddits(tx, subreddit_name):
    query = (
        "MATCH (s:Subreddit {name: $subreddit_name})-[r:REFERENCES]->(linked:Subreddit) "
        "RETURN linked.name AS linked_subreddit, r.node_type AS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





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.



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.

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

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

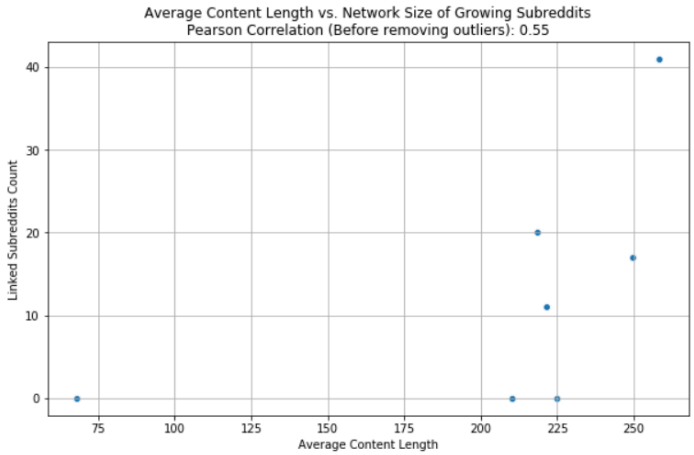
Graph Query

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.

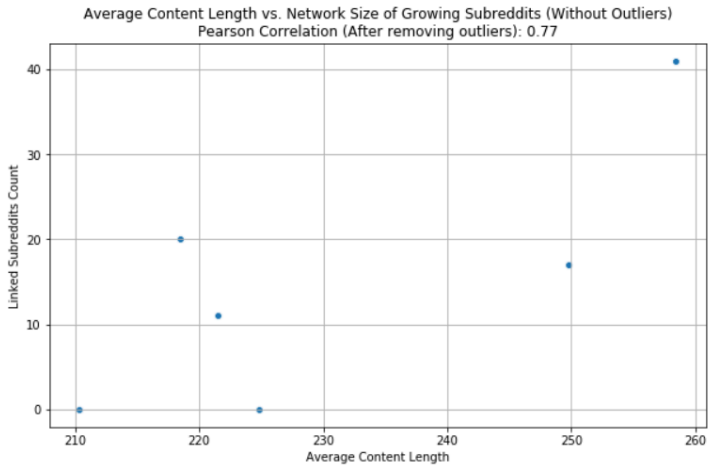
```python
# Function to find the number of linked subreddits for each growing subreddit
def find_linked_subreddits_count(tx, subreddit_name):
    query = (
        "MATCH (s:Subreddit {name: $subreddit_name})-[:REFERENCES]->(linked:Subreddit) "
        "RETURN COUNT(linked) AS linked_subreddit_count"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

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.

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

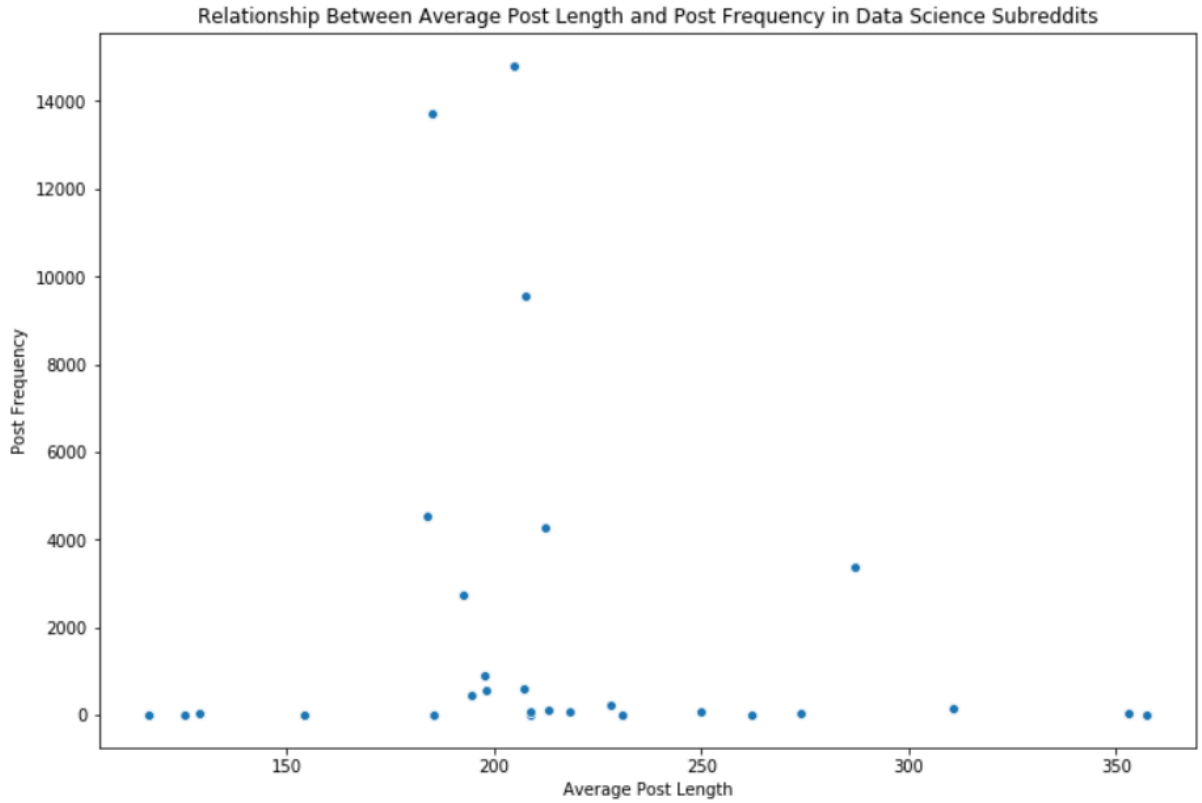
```
# Function to find newer data science subreddits influenced by the oldest ones
def get_influenced_subreddits(tx, old_subreddits):
    query = """
    MATCH (old:Subreddit)-[:REFERENCES]->(new:Subreddit)
    WHERE old.name IN $old_subreddits
    RETURN new.name AS subreddit, COUNT(*) AS influence_count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

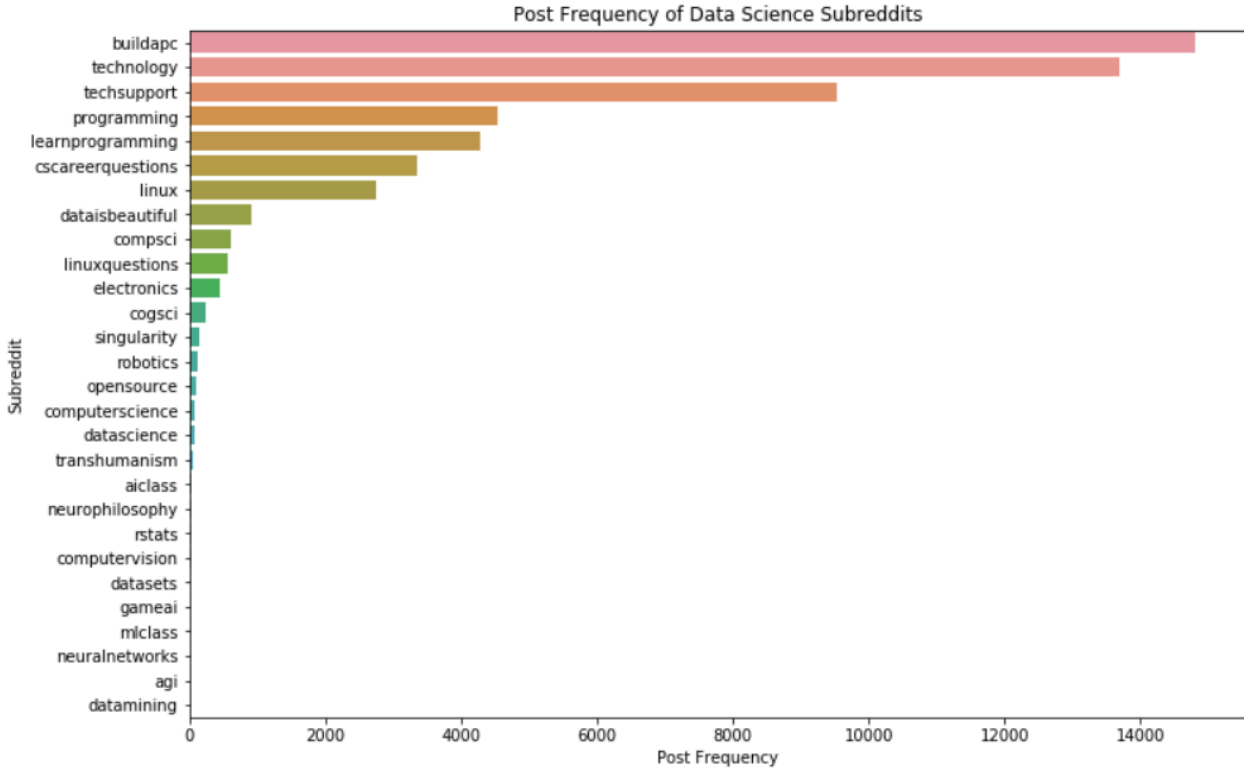
Query MongoDB: collection to calculate the average post length and post frequency for the newer data science subreddits that are influenced by the top 5 oldest subreddits

```
# Query to calculate the average post length and frequency for the influenced subreddits
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

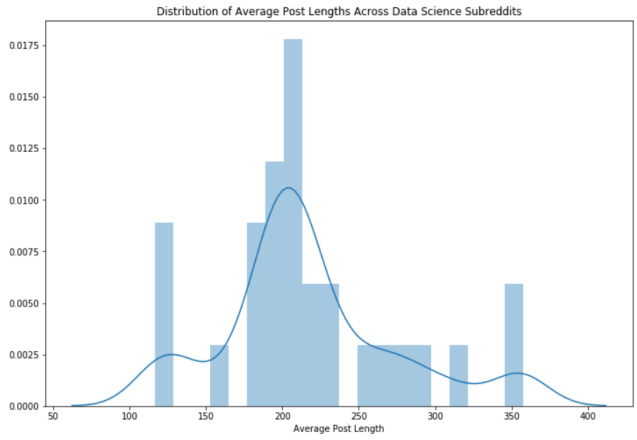


Subreddits with medium average post lengths tend to have higher post frequencies. This could indicate that communities with more in-depth discussions are also more active, and that active communities tend to normalize their post length on average.

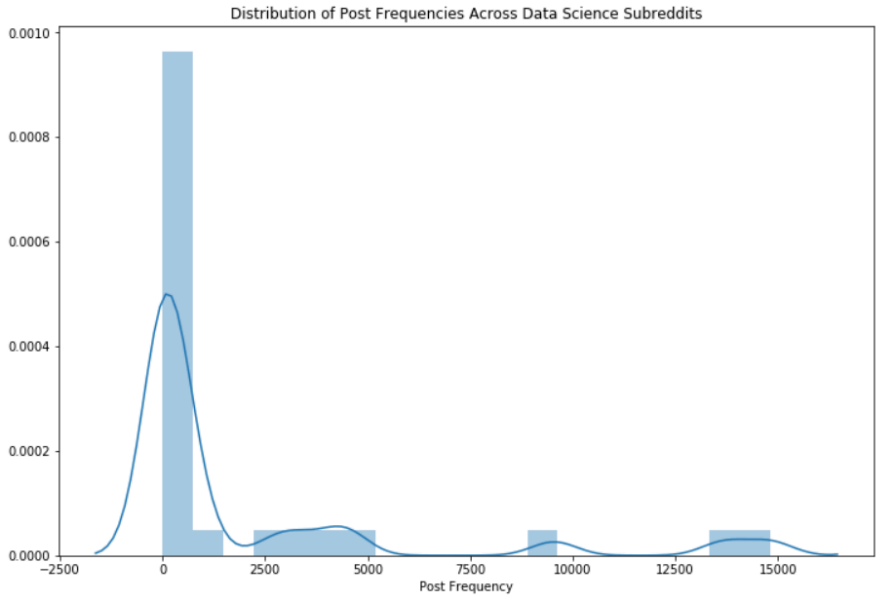


The bar chart ranks subreddits by post frequency and highlights the most active communities. There are vibrant hubs for data science related discussions.

The distribution of average post length 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.



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, andhiring.

```
# 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

Graph Query

Examine the links between career-focused subreddits and data subreddits in a Neo4jgraph database. Finds connections where a career-focused subreddit references and data subreddit containing the term 'data'.

```
# 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
7	dataengineering	dataisbeautiful	1
8	datasets	dataisbeautiful	1
9	datasets	datamining	1
10	artificial	datamining	1

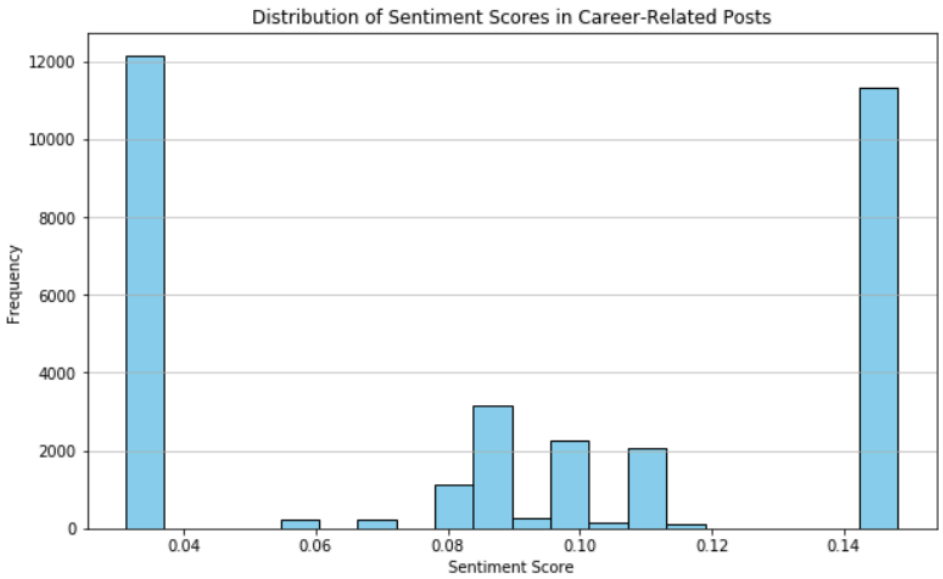
Sentiment Analysis

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

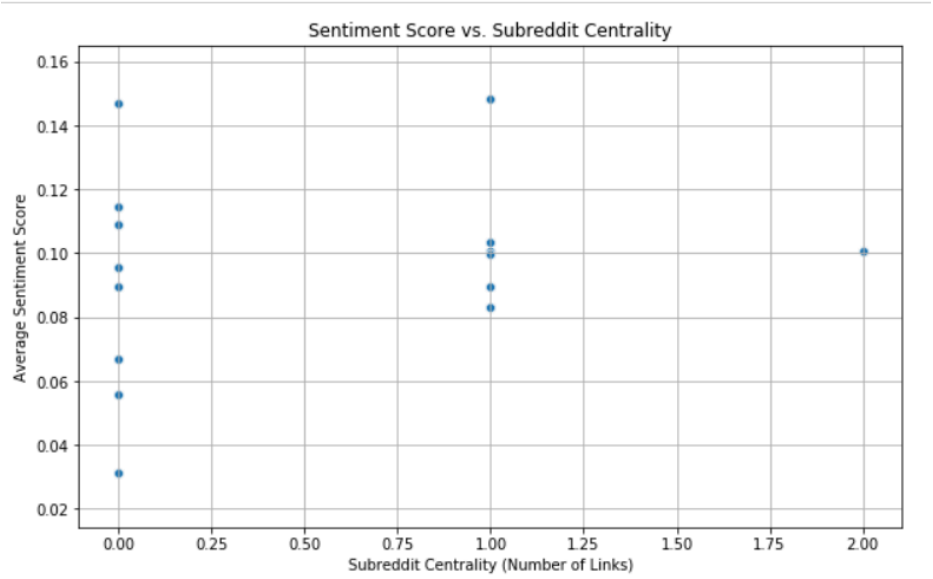
The code uses the TextBlob library to perform sentiment analysis on the content of each post. Sentiment is calculated using the polarity score,which represents the positivity or negativity of the text. For each subreddit, the average sentiment score is calculated.

```python
# 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

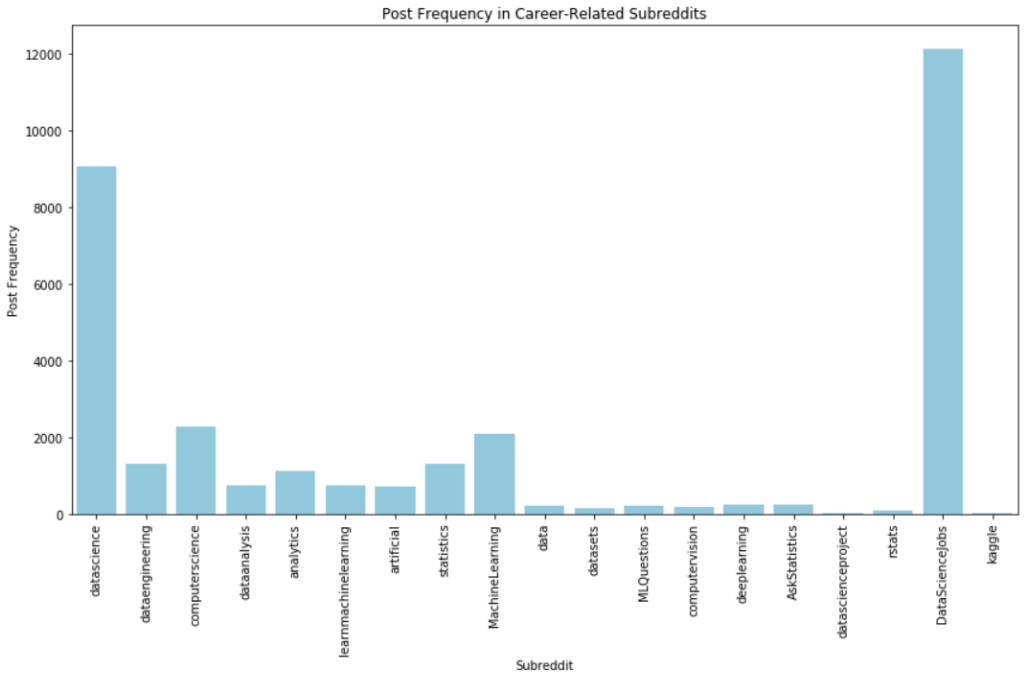


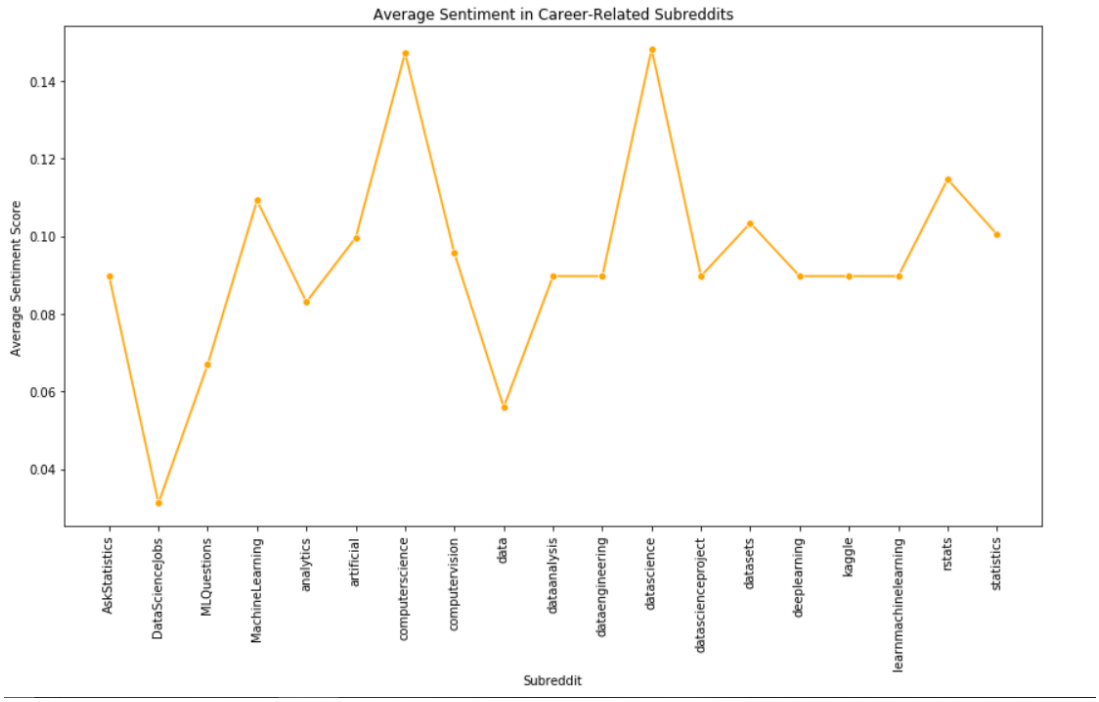
histogram of sentiment scores in career-related posts suggests a bimodal distribution, with a concentration of sentiment scores towards both ends.



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

Certain subreddits are hotspots for career-related discussions.





frequency of posts does not appear to be correlated with sentiment. This highlights the complexity of factors that contribute to the emotional tone of discussions, beyond just the quantity of conversation.

Key Findings

- **Growth and Connectivity:** We discovered a strong positive correlation between the growth of subreddits and their interconnectedness, indicated by the types of links they shared with other subreddits.

- **Content Volume and Network Size:** Our analysis revealed that subreddits with more extended discussions tend to have broader networks, suggested by an increased Pearson correlation coefficient after removing outliers.

- **Influence of Oldest Subreddits:** We analyzed how the oldest data science subreddits have influenced the development of newer ones. Our findings showed vibrant hubs of discussions with a moderate post length, suggesting that in-depth conversations could drive higher community activity.

Limitations

- **Engagement Data:** A significant constraint was the absence of engagement data in our MongoDB dataset, which restricted our ability to analyze user interactions comprehensively.

- **Time Constraints:** Due to the project's scope and time limitations, our analysis was primarily quantitative. We did not delve into the qualitative aspects of language changes over time.

- **Data Limitations:** Our data focused on data science-related subreddits, limiting the generalizability of our findings across the broader Reddit landscape.

- **Failed Queries:** Certain queries, like analyzing the influence of authors across subreddits or topic shifts over time, were not feasible due to data constraints or exceeded our project timeline.

References, Data 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

Future Work

- **Qualitative Analysis**: Future studies should include qualitative analyses, such as language sentiment over time, to provide a more nuanced understanding of community discussions.

- **Engagement Metrics**: Incorporating engagement metrics like upvotes, comments, and shares could offer deeper insights into user behavior and content popularity.

- **Broader Data Scope**: Expanding the dataset to include a wider array of subreddits could help validate our findings across different communities on Reddit.