```python
import praw
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import networkx as nx
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import datetime

# create a Reddit instance
reddit = praw.Reddit(client_id='5qcq6jExlBYGwe6mIZARSA',
                     client_secret='CRPMFHt6cATTQRl7KczNtovv9YJ4JA',
                     user_agent='Scraper 1.0 by /u/Sahan_-9830')

# specify the subreddit you want to retrieve posts from
""" headlines = set()
for submission in reddit.subreddit('silicon valley bank').hot(limit=None):
    print(submission.title)
    print(submission.id)
    print(submission.author)
    print(submission.created_utc)
    print(submission.score)
    print(submission.upvote_ratio)
    print(submission.url)
    break
    headlines.add(submission.title)
print(len(headlines)) """

# specify the subreddits you want to retrieve posts from
subreddits = ['all', 'startups', 'technology']

# specify the search queries
queries = ['silicon valley bank', 'svb', 'svb collapse','svb investment','svb startup','svb bankruptcy','silicon valley bank bankruptcy','svb bankrupt']

# create an empty list to store the data
data = []
comment = []

# loop through subreddits and queries and append data to list
for subreddit_name in subreddits:
    subreddit = reddit.subreddit(subreddit_name)
    for query in queries:
        search_results = subreddit.search(query)
        for post in search_results:
            data.append([post.title, post.selftext, post.score, post.num_comments, post.upvote_ratio, post.id, post.author.name if post.author else None,post.url,post.created_utc,post
            post.comments.replace_more(limit=100)
            for comments in post.comments:
                comment.append([comments.body, comments.score, comments.author.name if comments.author else None, comments.created_utc,post.title, post.id])

# create a pandas DataFrame from the data list
df = pd.DataFrame(data, columns=['title', 'body', 'score', 'num_comments', 'upvote_ratio', 'id', 'author','url','date','flair'])
df2 = pd.DataFrame(comment,columns=['comment_body','comment_score','comment_author','comment_utc','post_title','post_id'])
# print the DataFrame
print(df)
print(df2)

df2.to_csv('Comments_Data.csv', index=False)


# Download the VADER lexicon if needed
nltk.download('vader_lexicon')

# Create an instance of the SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()

# Define a function to classify sentiment
def classify_sentiment(title):
    # Apply VADER to get the sentiment scores
    sentiment_scores = sid.polarity_scores(title)
    # Extract the compound score
    compound_score = sentiment_scores['compound']
    # Classify as positive, negative, or neutral based on the compound score
    if compound_score > 0.05:
        return 'positive'
    elif compound_score < -0.05:
        return 'negative'
    else:
        return 'neutral'

# Apply the function to the 'title' column of the DataFrame
df['sentiment'] = df['title'].apply(classify_sentiment)

print(df)
df.to_csv('Reddit_Data.csv', index=False)


sns.set(style="darkgrid")

# create the countplot
sns.countplot(data=df, x='sentiment')

# set the title and labels for the plot
plt.title('Sentiment Analysis of Reddit Posts')
plt.xlabel('Sentiment')
plt.ylabel('Count')

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