

```
In [42]: import prawcore
import praw
import requests
import time
import pandas as pd
from datetime import datetime
import importlib.util
import requests
from datetime import datetime
import matplotlib.pyplot as plt
from textblob import TextBlob
import seaborn as sns
import scipy.stats as stats
from sklearn.feature_extraction.text import CountVectorizer
from nltk.corpus import stopwords
import string
import nltk
# nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\hanoz\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\stopwords.zip.
```

Out[42]: True

```
In [2]: # Load config module
config_path = "C:/Users/hanoz/OneDrive/Documents/github_projects/us_elections_senti
spec = importlib.util.spec_from_file_location("config", config_path)
config = importlib.util.module_from_spec(spec)
spec.loader.exec_module(config)
```

```
In [3]: # Access Reddit credentials
CLIENT_ID = config.CLIENT_ID
CLIENT_SECRET = config.CLIENT_SECRET
USER_AGENT = config.USER_AGENT
```

```
In [4]: # Set up Reddit client
reddit = praw.Reddit(client_id=CLIENT_ID, client_secret=CLIENT_SECRET, user_agent=U
```

```
In [5]: # Initialize an empty DataFrame
df_combined = pd.DataFrame()
```

```
In [6]: # Updated function to fetch all recent posts from a specific subreddit
def fetch_subreddit_posts(subreddit_name, limit=1000, before_timestamp=None):
    subreddit = reddit.subreddit(subreddit_name)
    posts = []

    # Use params with `before` timestamp to get older posts if available
    params = {'before': before_timestamp} if before_timestamp else {}

    # Fetch the latest posts without filtering by a search term
    for submission in subreddit.new(limit=limit):
        post_date = datetime.fromtimestamp(submission.created_utc)
        posts.append({
```

```

        "id": submission.id,
        "title": submission.title,
        "created_utc": post_date,
        "score": submission.score,
        "num_comments": submission.num_comments,
        "selftext": submission.selftext,
        "subreddit": subreddit_name
    })

    return pd.DataFrame(posts)

```

```

In [7]: # Set your target for the number of posts
target_post_count = 2000
current_post_count = 0

```

```

In [8]: # Track the timestamp to continue fetching older posts
last_timestamp = None

```

```

In [9]: # Subreddits you want to scrape (you can add more if desired)
subreddits = ["politics", "PoliticalDiscussion", "Ask_Politics", "AskElectoralColle

```

```

In [10]: while current_post_count < target_post_count:
    for subreddit_name in subreddits:
        try:
            # Attempt to fetch posts from each subreddit
            df_new = fetch_subreddit_posts(subreddit_name, limit=1000, before_times

            # Check if data is returned
            if df_new.empty:
                print(f"No more posts found for {subreddit_name}.")
                continue

            # Update timestamp to the oldest post
            last_timestamp = int(df_new["created_utc"].min().timestamp())

            # Append to the combined DataFrame, avoiding duplicates
            df_combined = pd.concat([df_combined, df_new]).drop_duplicates(subset="

            # Update the post count
            current_post_count = len(df_combined)
            print(f"Total posts collected so far: {current_post_count}")

            # Break if the target is reached
            if current_post_count >= target_post_count:
                break

        except prawcore.exceptions.Redirect:
            print(f"Subreddit '{subreddit_name}' caused a redirect and may be priva
            continue # Skip to the next subreddit

        # Pause to respect rate limits
        time.sleep(5)

    print("Data collection complete.")

```

Total posts collected so far: 990
 Total posts collected so far: 1470
 Total posts collected so far: 2061
 Data collection complete.

In [11]: `df_combined.head()`

Out[11]:

	id	title	created_utc	score	num_comments	selftext	subreddit
0	1gn0lr5	His Father Is Leaving Office. Is Hunter Biden'...	2024-11-08 22:05:12	1	3		politics
1	1gn0ksh	Pelosi blames Biden for election loss as finge...	2024-11-08 22:03:44	5	3		politics
2	1gn0h7j	Democratic Congresswoman Marie Gluesenkamp Pér...	2024-11-08 21:58:16	4	4		politics
3	1gn0ame	Elon Musk joined Trump-Zelensky call amid conc...	2024-11-08 21:48:01	11	25		politics
4	1gn09ww	Democrats Hoped the Bros Wouldn't Show. But Th...	2024-11-08 21:46:55	0	29		politics

In [12]: *# Define a function to calculate sentiment polarity*

```
def analyze_sentiment(text):
    blob = TextBlob(text)
    return blob.sentiment.polarity
```

In [13]: *# Apply sentiment analysis to the 'title' column*

```
df_combined["sentiment"] = df_combined["title"].apply(analyze_sentiment)
```

In [14]: *# Display the DataFrame with the sentiment scores*

```
df_combined[["title", "sentiment"]].head()
```

Out[14]:

	title	sentiment
0	His Father Is Leaving Office. Is Hunter Biden'...	0.00
1	Pelosi blames Biden for election loss as finge...	0.00
2	Democratic Congresswoman Marie Gluesenkamp Pér...	0.15
3	Elon Musk joined Trump-Zelensky call amid conc...	0.00
4	Democrats Hoped the Bros Wouldn't Show. But Th...	0.00

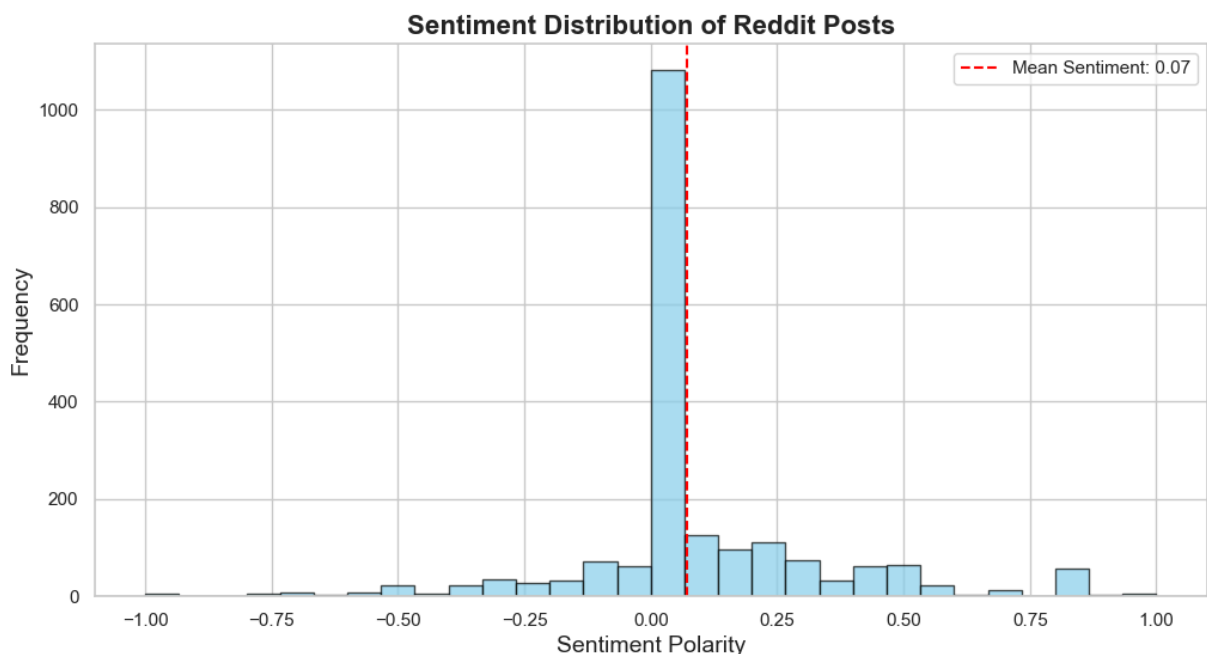
```
In [15]: # Set the style and palette using Seaborn for enhanced aesthetics
sns.set(style="whitegrid")

plt.figure(figsize=(12, 6))
plt.hist(df_combined["sentiment"], bins=30, color="skyblue", edgecolor="black", alp
plt.axvline(df_combined["sentiment"].mean(), color="red", linestyle="dashed", linew

# Title and Labels
plt.title("Sentiment Distribution of Reddit Posts", fontsize=16, fontweight='bold')
plt.xlabel("Sentiment Polarity", fontsize=14)
plt.ylabel("Frequency", fontsize=14)

plt.legend()

plt.show()
```



```
In [16]: # Calculate various statistics for sentiment data
mean_sentiment = df_combined["sentiment"].mean()
median_sentiment = df_combined["sentiment"].median()
std_sentiment = df_combined["sentiment"].std()
min_sentiment = df_combined["sentiment"].min()
max_sentiment = df_combined["sentiment"].max()
skewness_sentiment = stats.skew(df_combined["sentiment"])
kurtosis_sentiment = stats.kurtosis(df_combined["sentiment"])

# Display the statistics
print(f"Mean Sentiment: {mean_sentiment:.2f}")
print(f"Median Sentiment: {median_sentiment:.2f}")
print(f"Standard Deviation of Sentiment: {std_sentiment:.2f}")
print(f"Minimum Sentiment: {min_sentiment:.2f}")
print(f"Maximum Sentiment: {max_sentiment:.2f}")
print(f"Skewness of Sentiment: {skewness_sentiment:.2f}")
print(f"Kurtosis of Sentiment(Measures the 'tailedness' of the distribution. High k
```

Mean Sentiment: 0.07

Median Sentiment: 0.00

Standard Deviation of Sentiment: 0.25

Minimum Sentiment: -1.00

Maximum Sentiment: 1.00

Skewness of Sentiment: 0.48

Kurtosis of Sentiment(Measures the 'tailedness' of the distribution. High kurtosis indicates more outliers, while low kurtosis indicates fewer outliers than a normal distribution.): 3.17

```
In [17]: # Ensure the date column is in datetime format and filter data for 2024
df_combined["date"] = pd.to_datetime(df_combined["created_utc"]).dt.date
df_combined_2024 = df_combined[df_combined["date"].apply(lambda x: x.year) == 2024]
```

```
In [18]: # Group by date and calculate the mean sentiment for each day in 2024
daily_sentiment_2024 = df_combined_2024.groupby("date")["sentiment"].mean().reset_index()
```

```
In [19]: plt.figure(figsize=(14, 7))
plt.plot(daily_sentiment_2024["date"], daily_sentiment_2024["sentiment"], color="skyblue",
         marker="o", linestyle="solid", linewidth=1.5)

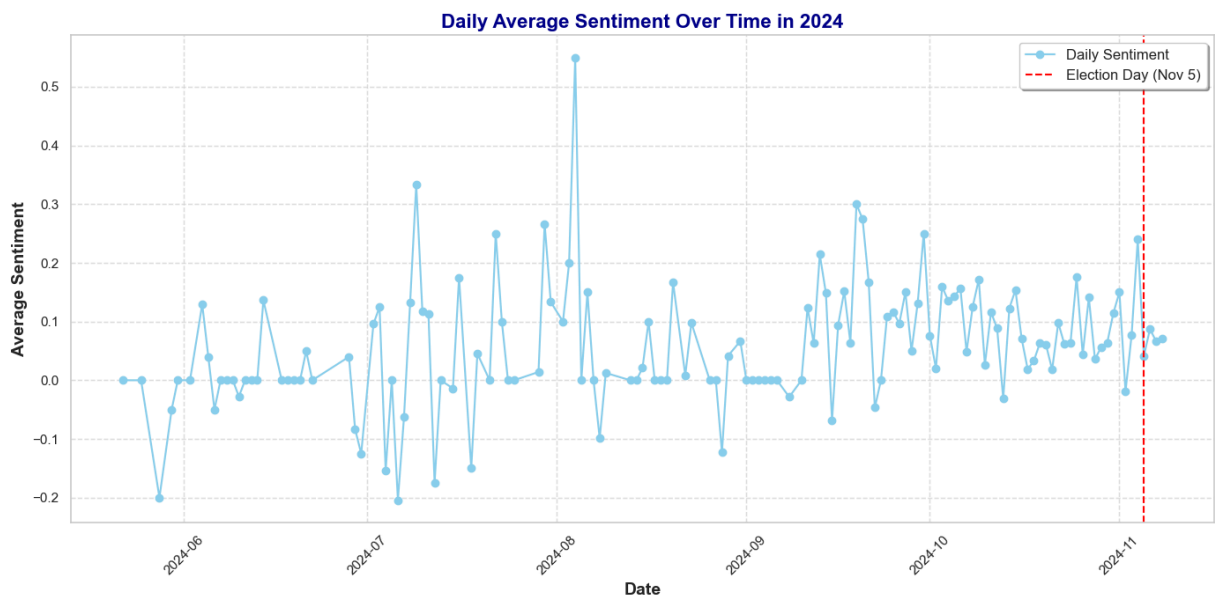
# Highlighting specific event
election_day = pd.to_datetime("2024-11-05")
plt.axvline(election_day, color="red", linestyle="--", linewidth=1.5, label="Election Day")

# Title and Labels with enhanced styling
plt.title("Daily Average Sentiment Over Time in 2024", fontsize=16, fontweight="bold")
plt.xlabel("Date", fontsize=14, fontweight="bold")
plt.ylabel("Average Sentiment", fontsize=14, fontweight="bold")

plt.xticks(rotation=45)
plt.grid(True, linestyle="--", alpha=0.6)

plt.legend(loc="upper right", frameon=True, shadow=True, fontsize=12)

plt.tight_layout()
plt.show()
```



```

In [44]: # Define stop words and punctuation
stop_words = set(stopwords.words("english"))
punctuation = set(string.punctuation)

In [46]: # Function to clean text by removing stopwords and punctuation
def clean_text(text):
    words = text.lower().split()
    return " ".join(word for word in words if word not in stop_words and word not in punctuation)

In [50]: # Apply cleaning function to the 'title' column using .loc to avoid the SettingWithCopyError
df_combined_2024.loc[:, "cleaned_title"] = df_combined_2024["title"].apply(clean_text)

In [52]: # Separate positive and negative sentiment posts
positive_posts = df_combined_2024[df_combined_2024["sentiment"] > 0]
negative_posts = df_combined_2024[df_combined_2024["sentiment"] < 0]

In [54]: # Initialize CountVectorizer to find top keywords
vectorizer = CountVectorizer(max_features=5)

In [56]: # Find top 5 keywords for positive posts
positive_vectorized = vectorizer.fit_transform(positive_posts["cleaned_title"])
positive_keywords = vectorizer.get_feature_names_out()

In [58]: # Find top 5 keywords for negative posts
negative_vectorized = vectorizer.fit_transform(negative_posts["cleaned_title"])
negative_keywords = vectorizer.get_feature_names_out()

In [60]: # Display the results
print("Top 5 Keywords in Positive Sentiments:", positive_keywords)
print("Top 5 Keywords in Negative Sentiments:", negative_keywords)

Top 5 Keywords in Positive Sentiments: ['election' 'harris' 'new' 'trump' 'win']
Top 5 Keywords in Negative Sentiments: ['democrats' 'election' 'harris' 'kamala' 'trump']

In [62]: # Define keywords for each candidate
harris_keywords = ["Harris", "Kamala Harris", "Kamala"]
trump_keywords = ["Trump", "Donald Trump", "Donald"]

In [64]: # Filter posts mentioning Biden
harris_posts = df_combined_2024[df_combined_2024["title"].str.contains('|'.join(harris_keywords))]

# Filter posts mentioning Trump
trump_posts = df_combined_2024[df_combined_2024["title"].str.contains('|'.join(trump_keywords))]

```

```
In [66]: # Calculate sentiment statistics for each candidate
harris_sentiment_mean = harris_posts["sentiment"].mean()
harris_sentiment_median = harris_posts["sentiment"].median()
harris_sentiment_std = harris_posts["sentiment"].std()

trump_sentiment_mean = trump_posts["sentiment"].mean()
trump_sentiment_median = trump_posts["sentiment"].median()
trump_sentiment_std = trump_posts["sentiment"].std()
```

```
In [68]: # Display statistics
print("Harris Sentiment - Mean:", harris_sentiment_mean, "Median:", harris_sentiment_median, "Standard Deviation:", harris_sentiment_std)
print("Trump Sentiment - Mean:", trump_sentiment_mean, "Median:", trump_sentiment_median, "Standard Deviation:", trump_sentiment_std)
```

Harris Sentiment - Mean: 0.055305731886449376 Median: 0.0 Standard Deviation: 0.2787303960513446

Trump Sentiment - Mean: 0.11384220160592995 Median: 0.0 Standard Deviation: 0.2909449431368743

```
In [78]: plt.figure(figsize=(12, 7))

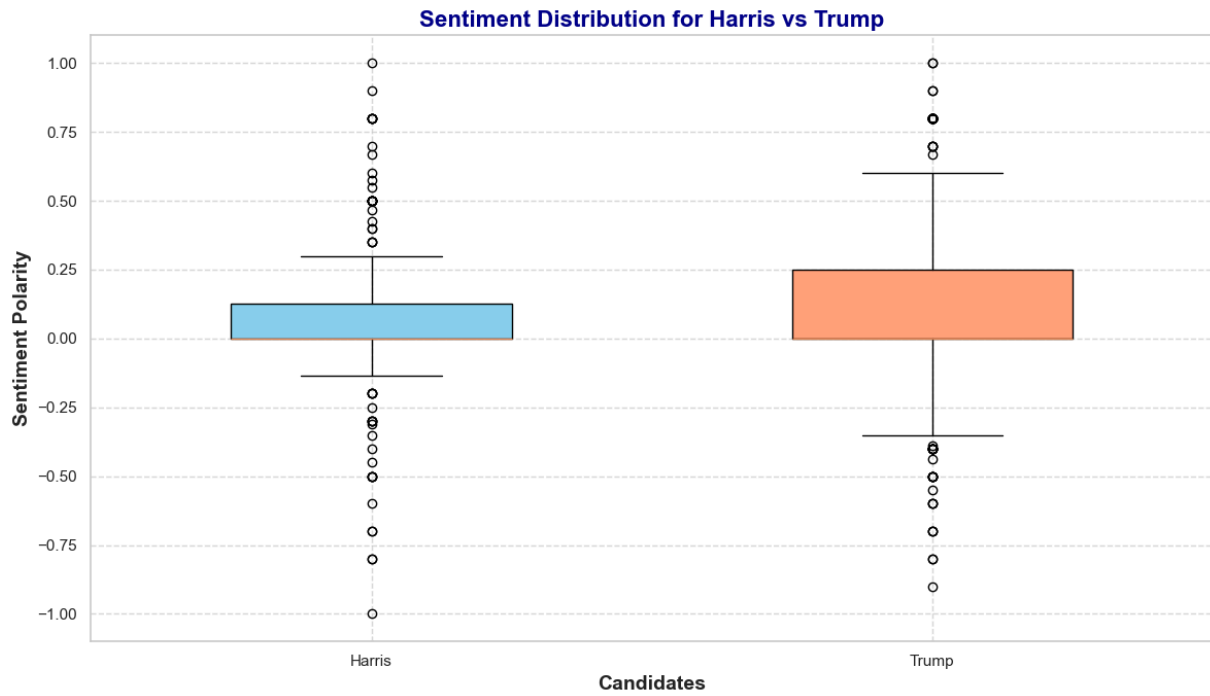
box = plt.boxplot(
    [harris_posts["sentiment"].dropna(), trump_posts["sentiment"].dropna()],
    patch_artist=True, # Fill the boxes with color
    tick_labels=["Harris", "Trump"],
    widths=0.5,
)

colors = ["#87CEEB", "#FFA07A"]
for patch, color in zip(box['boxes'], colors):
    patch.set_facecolor(color)

plt.title("Sentiment Distribution for Harris vs Trump", fontsize=16, fontweight="bold")
plt.ylabel("Sentiment Polarity", fontsize=14, fontweight="bold")
plt.xlabel("Candidates", fontsize=14, fontweight="bold")

plt.grid(True, linestyle="--", alpha=0.7)

plt.tight_layout()
plt.show()
```



```
In [86]: df_combined_2024.to_csv("reddit_sentiment_analysis_2024.csv", index=False)
```

```
In [88]: df_check = pd.read_csv("reddit_sentiment_analysis_2024.csv")
print(df_check.head())
```

	id	title \
0	1gn0lr5	His Father Is Leaving Office. Is Hunter Biden'...
1	1gn0ksh	Pelosi blames Biden for election loss as finge...
2	1gn0h7j	Democratic Congresswoman Marie Gluesenkamp Pér...
3	1gn0ame	Elon Musk joined Trump-Zelensky call amid conc...
4	1gn09ww	Democrats Hoped the Bros Wouldn't Show. But Th...

	created_utc	score	num_comments	selftext	subreddit	sentiment \
0	2024-11-08 22:05:12	1	3	NaN	politics	0.00
1	2024-11-08 22:03:44	5	3	NaN	politics	0.00
2	2024-11-08 21:58:16	4	4	NaN	politics	0.15
3	2024-11-08 21:48:01	11	25	NaN	politics	0.00
4	2024-11-08 21:46:55	0	29	NaN	politics	0.00

	date	cleaned_title
0	2024-11-08	father leaving office. hunter biden's art mark...
1	2024-11-08	pelosi blames biden election loss finger point...
2	2024-11-08	democratic congresswoman marie gluesenkamp pér...
3	2024-11-08	elon musk joined trump-zelensky call amid conc...
4	2024-11-08	democrats hoped bros show. did.

```
In [ ]:
```