Final Report – Social Media Sentiment Analysis of POTUS

Student Name: Alexandre Cabral and Jordan Schmutzler
Course: PS25 - SE 02 Big Data & Analytics
Date: 2025-07-04

---

1. Introduction

This project is a sentiment analysis of the brand "Trump" in the context of a specific international crisis: a U.S. military attack on Iran ordered by President Trump. The study focuses on public reaction as expressed in a single Reddit thread from the subreddit r/AskReddit titled *"Trump bombs Iran. What do you think this will lead to?"* (link). It's about understanding brand dynamics under pressure.

The analysis involved building a full data pipeline: collecting comments with Python, cleaning and processing them via Hadoop MapReduce, storing results in PostgreSQL, and performing sentiment classification in Python. Results give us insight into the polarity and emotional weight behind the reactions.

---

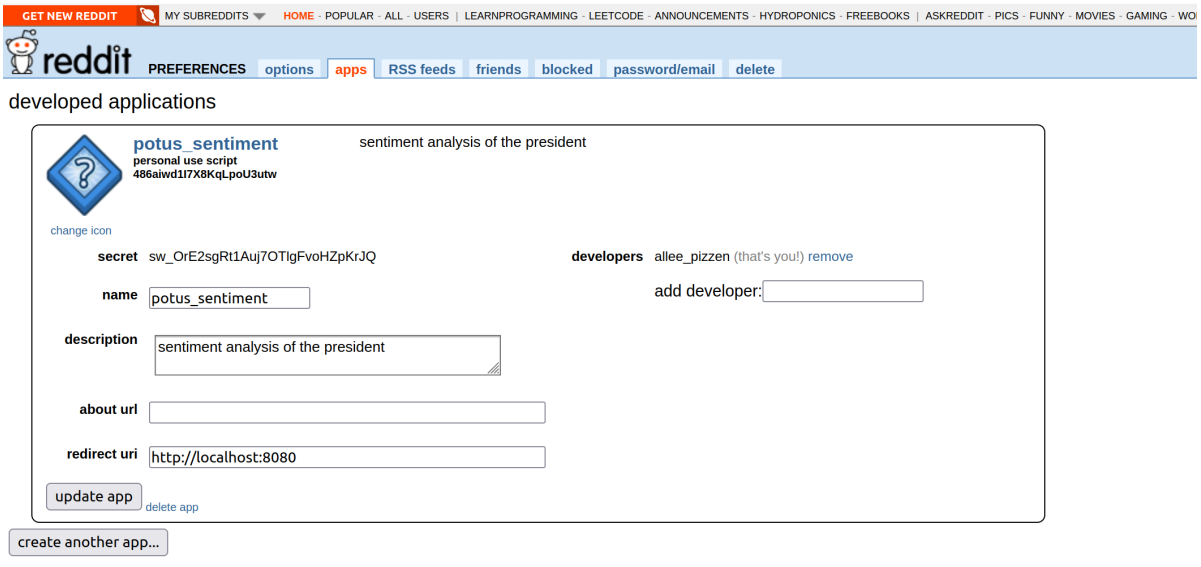
2. Problem Definition & Data Collection

2.1 Problem Statement

To analyze how a U.S. military strike on Iran ordered by President Trump affected public perception of his persona as a brand, by performing sentiment analysis on user comments from a specific Reddit thread in r/AskReddit.

2.2 Data Acquisition

2.2.1 Reddit App creation - secret key



2.2.2 Reddit API

- **Search Term:** `Trump`
- **Posts Collected:** Approx. 7000 posts/comments retrieved using search queries on Reddit
- **Tool Used:** Python Reddit API Wrapper (PRAW) and direct JSON data storage.

```python
import praw
import json
from datetime import datetime
from dotenv import load_dotenv
import os
import argparse

# Load credentials from .env
load_dotenv()
client_id = os.getenv("CLIENT_ID")
client_secret = os.getenv("SECRET")
user_agent = os.getenv("AGENT")

# Connect to Reddit API
reddit = praw.Reddit(client_id=client_id,
                     client_secret=client_secret,
                     user_agent=user_agent)

# Parse arguments
parser = argparse.ArgumentParser()
parser.add_argument("--url", required=True, help="Reddit post URL")
args = parser.parse_args()

submission = reddit.submission(url=args.url)
submission.comments.replace_more(limit=None)
```

,

```python

comments = []

def extract_comments(comment_list, parent=None):
    for comment in comment_list:
        comments.append({
            "id": comment.id,
            "parent_id": parent,
            "author": str(comment.author),
            "score": comment.score,
            "body": comment.body,
            "created_utc": comment.created_utc
        })
        extract_comments(comment.replies, parent=comment.id)

extract_comments(submission.comments)

# Save to JSON
os.makedirs("data/raw", exist_ok=True)
post_id = submission.id
filename = f"data/raw/reddit_{post_id}_comments.json"
with open(filename, "w") as f:
    json.dump(comments, f, indent=2)

print(f" Saved {len(comments)} comments to {filename}")
```

- Data collected and stored in structured JSON format
- File stored locally as `reddit_trump_comments.json`

3. Data Engineering Environment Setup

3.1 Technical Configuration

- **OS:** Ubuntu 22.04
- **Java Version:** OpenJDK 11 (JAVA 11) selected and set correctly using `JAVA_HOME` and `update-alternatives`.
- **Hadoop Version:** 3.3.6
- **Python Environment:** Virtualenv with `nltk`, `TextBlob`, `VADER`, `psycopg2`, and other required libraries
- **Shell:** zsh configured via `.zshrc`
- **Java Version:**
- **Hadoop Version:** 3.3.6 installed and configured.
- **Hadoop Services Running:** `NameNode`, `DataNode`, `ResourceManager`, `NodeManager`, `SecondaryNameNode`.

3.2 Challenges Resolved

- Incompatible Java version (Java 24) was replaced with Java 17 to support Hadoop, then again replaced by Java 11 for compatibility reasons.
- Manually launched YARN components when automatic launch failed.

---

4. Data Preprocessing with MapReduce

4.1 Cleaning Goals

- Remove stopwords, URLs, punctuation, emojis
- Normalize to lowercase
- Remove duplicates and missing values

4.2 mapperflatter.py (Python)

scripts > mapperflatter.py > ...

```python
#!/usr/bin/env python3
#!/home/alex/UE-Germany/ps25_bigdata/Sentiment_Tiktok/.venv/bin/python

import nltk
nltk.download('stopwords')
import sys
import json
import re
from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))

def clean_text(text):
    text = text.lower()
    text = re.sub(r"http\S+", "", text)
    text = re.sub(r"[^\w\s]", "", text)
    return [word for word in text.split() if word not in stop_words]

def process(record):
    if 'body' in record and isinstance(record['body'], str):
        words = clean_text(record['body'])
        for word in words:
            print(f"{word}\t1")

try:
    content = sys.stdin.read()
    data = json.loads(content)
    if isinstance(data, list):
        for record in data:
            process(record)
    else:
        process(data)
except Exception:
    pass
```

This script is called "mapperflatter" because it not only maps over each object in the input array (processing each record individually), but also "flattens" the structure by emitting each word from the 'body' field as a separate output line. This flattening transforms nested or grouped data into a simple, line-by-line format suitable for further processing in data pipelines.

4.3 reducer.py (Python)

scripts > reducer.py > ...

```python
#!/usr/bin/env python3
#!/home/alex/UE-Germany/ps25_bigdata/Sentiment_Tiktok/.venv/bin/python

import sys

current_word = None
current_count = 0

for line in sys.stdin:
    word, count = line.strip().split('\t')
    count = int(count)

    if word == current_word:
        current_count += count
    else:
        if current_word is not None:
            print(f"{current_word}\t{current_count}")
        current_word = word
        current_count = count

# Don't forget the last word
if current_word is not None:
    print(f"{current_word}\t{current_count}")

```

4.4 Execution Command

```
~/UE-Germany/ps25_bigdata/Sentiment_Tiktok/scripts on ▫ Setup! ⌚ 14:24:59
$ hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-*.jar \
  -D mapreduce.framework.name=local \
  -D fs.defaultFS=file:/// \
  -files /home/alex/UE-
Germany/ps25_bigdata/Sentiment_Tiktok/scripts/mapperflatter.py,/home/alex/UE-
Germany/ps25_bigdata/Sentiment_Tiktok/scripts/reducer.py \
  -input /home/alex/UE-
Germany/ps25_bigdata/Sentiment_Tiktok/scripts/data/raw/reddit_trump_comments.json
\
  -output /home/alex/UE-
Germany/ps25_bigdata/Sentiment_Tiktok/scripts/output_local \
  -mapper mapperflatter.py \
  -reducer reducer.py \
  -cmdenv PYTHONIOENCODING=utf8 \
  -cmdenv PATH=/home/alex/UE-
Germany/ps25_bigdata/Sentiment_Tiktok/.venv/bin:/usr/bin:/bin
```

5. Data Integration with PostgreSQL

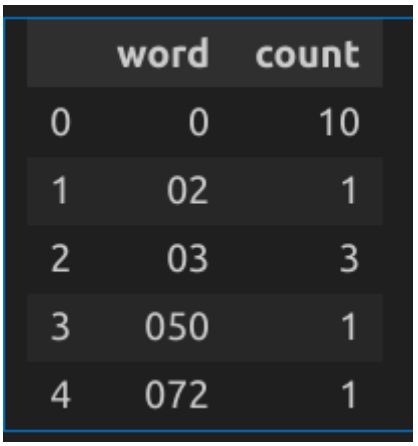
5.1 Transfer

- Cleaned output from Hadoop loaded into PostgreSQL

Postgresql in terminal:

```
$ sudo service postgresql start

~/UE-Germany/ps25_bigdata/Sentiment_Tiktok/scripts on  Setup! ⌚ 14:41:01
$ sudo -u postgres psql
could not change directory to "/home/alex/UE-Germany/ps25_bigdata/Sentiment_Tiktok/scripts": Permission denied
psql (14.18 (Ubuntu 14.18-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# CREATE DATABASE sentiment_analysis;
CREATE USER alex WITH PASSWORD '###########';
GRANT ALL PRIVILEGES ON DATABASE sentiment_analysis TO alex;
\q
CREATE DATABASE
CREATE ROLE
GRANT

~/UE-Germany/ps25_bigdata/Sentiment_Tiktok/scripts on  Setup! ⌚ 14:43:11
$ psql -U alex -d sentiment_analysis

psql (14.18 (Ubuntu 14.18-0ubuntu0.22.04.1))
Type "help" for help.

sentiment_analysis=> CREATE TABLE word_count (
    word TEXT PRIMARY KEY,
    count INTEGER
);
CREATE TABLE
sentiment_analysis=> \q
```

5.2 FINALLY, first output... (underwhelming)



	word	count
0	0	10
1	02	1
2	03	3
3	050	1
4	072	1

6. EDA (Notebook-Based)

The main exploration was performed in Jupyter Notebook

6.1 Tools Used

```
import pandas as pd
import psycopg2
import json
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
import re
from wordcloud import WordCloud
from nltk.sentiment import SentimentIntensityAnalyzer
import nltk
```

6.2 Simple Visualizations for exploratory data analysys

- **Main dataframe** (comments_df)

	author	body	created_utc	id	parent_id	score
0	GOD-PORING	July 4 heightened security at airports and bor...	2025-06-22 01:15:53	mz2we19	None	8054
1	the_replicator	Funny you mention that…\n \n[Republicans are c...	2025-06-22 02:48:21	mz3arti	mz2we19	3424
2	Octane14	only because they want to privatize it...	2025-06-22 05:45:10	mz3xuq8	mz3arti	4232
3	Ok-Driver-6277	I don't really understand why people aren't co...	2025-06-22 07:04:59	mz46i6h	mz3xuq8	2338
4	Aleashed	They don't care if there is another 9/11\n\nTh...	2025-06-22 10:45:55	mz4sqgt	mz46i6h	1075
...	...	...	...	...	...	...
6873	PippaTulip	Ah yes, bombing another country will keep us f...	2025-06-22 10:18:32	mz4pvih	mz4mnps	0
6874	Gold_Age_3768	lol	2025-06-22 11:21:23	mz4wta9	mz4pvih	1
6875	Grizzly_Addams	Pussy liberals pussing everywhere	2025-06-23 02:59:56	mz9kutq	None	-6
6876	EntreNous_2112	That's my man 🤘 Way to go!	2025-06-22 07:42:43	mz4ad42	None	-6
6877	RedOneRanger	If our commander in chief is doing it, he must...	2025-06-22 05:24:34	mz3vhsz	None	-12

6878 rows × 6 columns

- **Most active users, count of comments over time, avrg chars per comment**

	author	posts
0	None	163
1	Upstairs_Eagle_4780	57
2	Letterkenny-Wayne	26
3	StickShiftTudor	20
4	that_guy898	19
5	ContraryConman	13
6	MichiganPilotDaddy	13
7	scootiescoo	13
8	joe1max	12
9	HypnoToadVictim	12

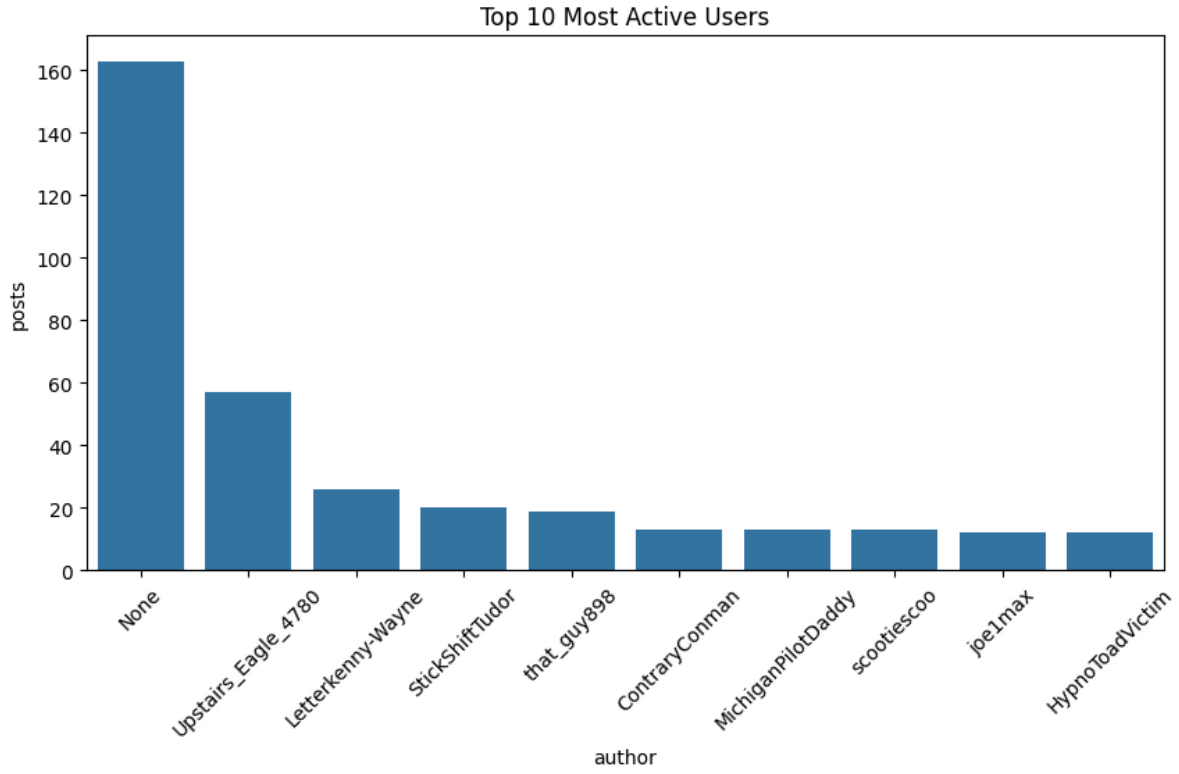
	day	count
0	2025-06-22	6518
1	2025-06-23	260
2	2025-06-24	52
3	2025-06-25	19
4	2025-06-26	8
5	2025-06-27	9
6	2025-06-28	9
7	2025-07-01	1
8	2025-07-02	1
9	2025-07-03	1

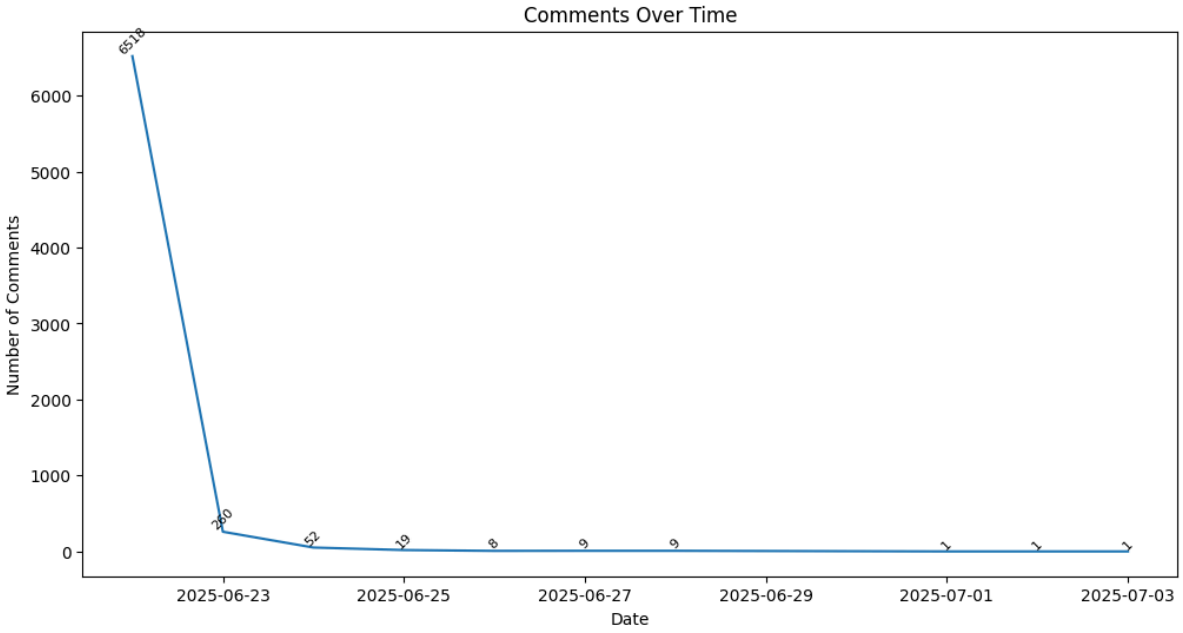
	avg
0	155.434574

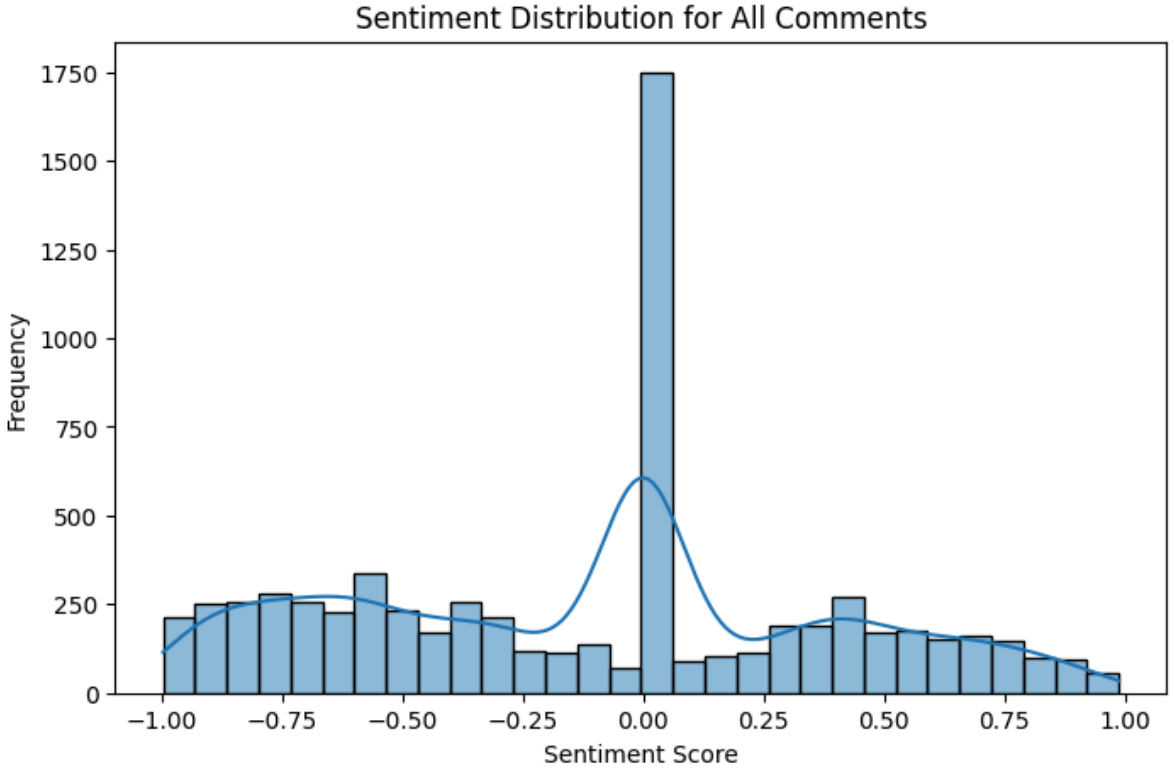
6.2 Sentiment Classification Logic

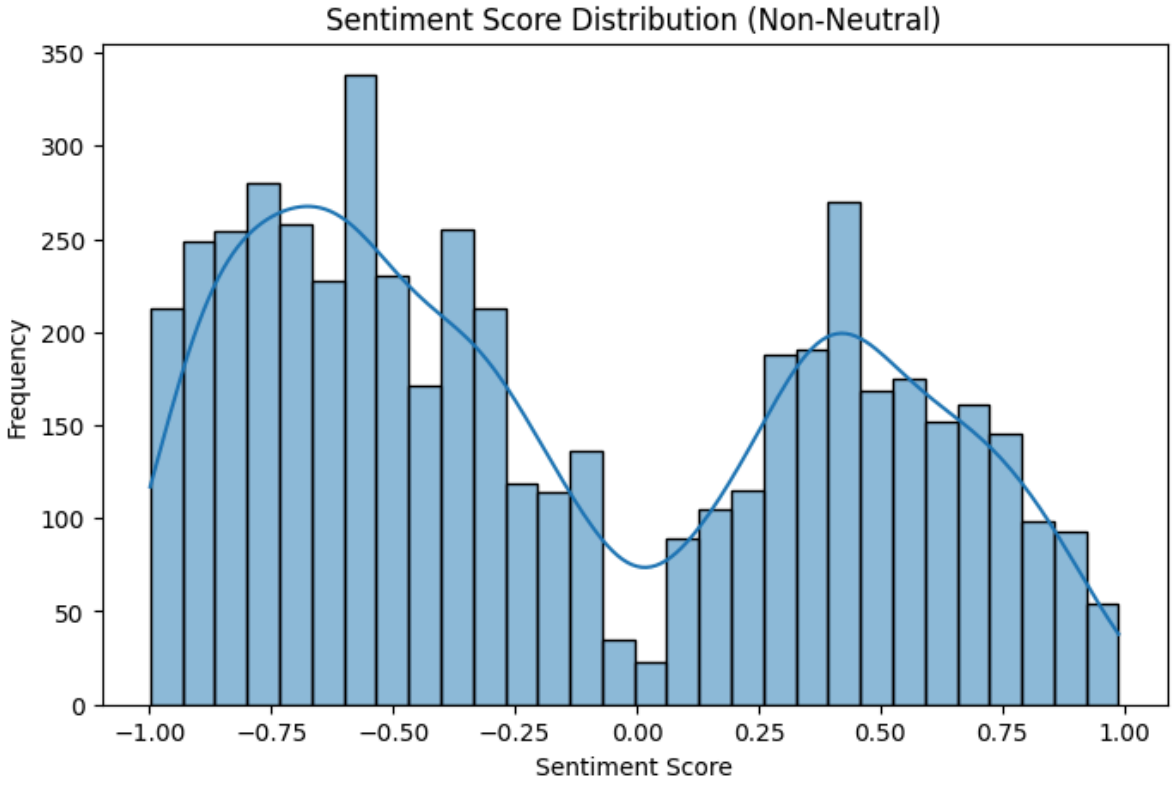
- Polarize text (scores from -1 to +1)
- Score each comment with compound sentiment value

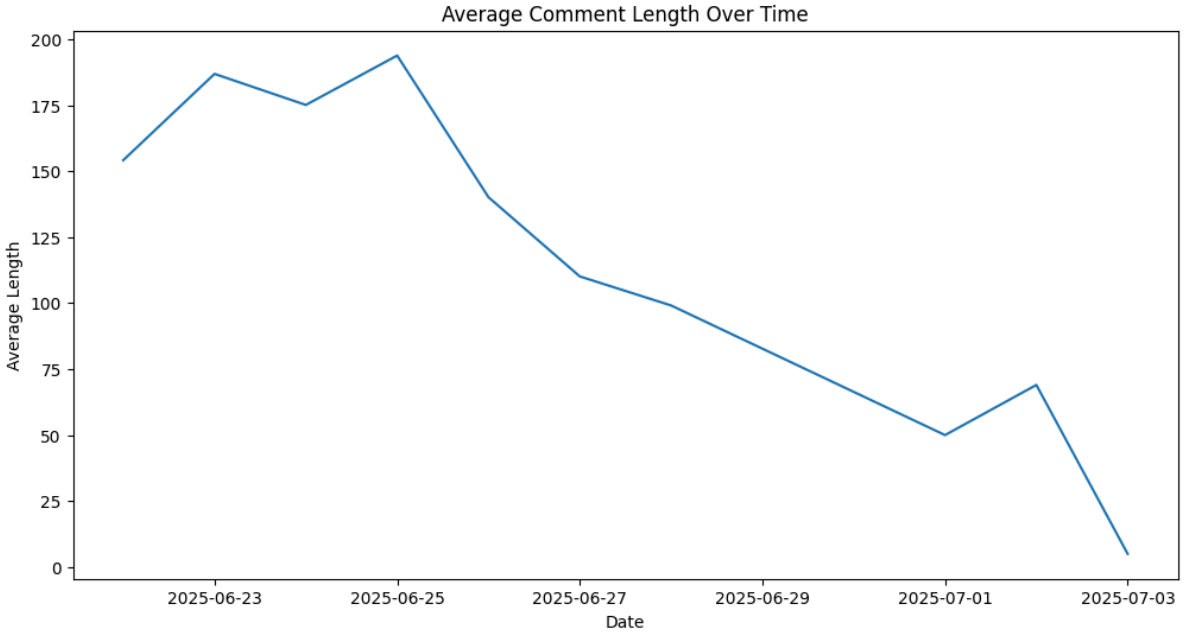
6.3 Findings




Comments Over Time
Number of Comments
Date
6518
260
52
19
8
9
9
1
1
1
0
1000
2000
3000
4000
5000
6000
2025-06-23
2025-06-25
2025-06-27
2025-06-29
2025-07-01
2025-07-03

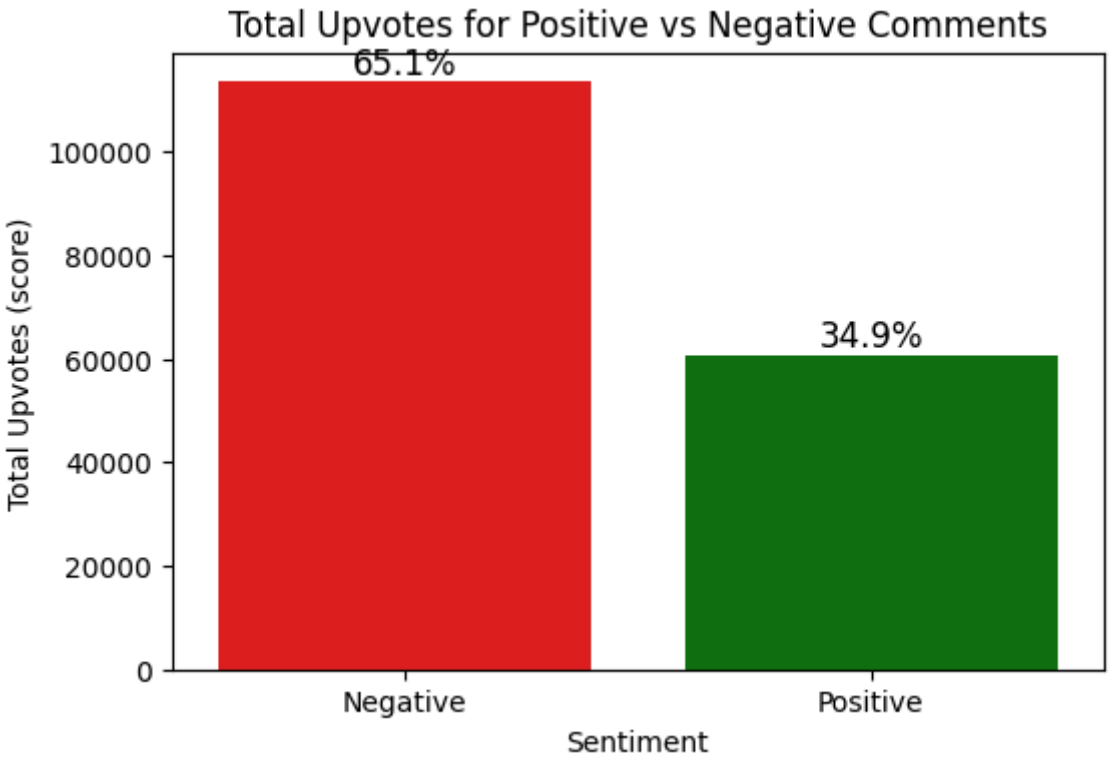

Sentiment Distribution for All Comments
Frequency
Sentiment Score
0
250
500
750
1000
1250
1500
1750
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00





Positive Comments Word Cloud

Negative Comments Word Cloud



8. Conclusion & Recommendations

At first glance, the sentiment distribution in the thread appeared balanced—many users voiced either support or disapproval of Trump's decision to bomb Iran. However, introducing the "score" parameter (upvotes) showed a more revealing picture: negative comments toward Trump consistently received the most support from the Reddit community. This suggests that, although users may have expressed a range of opinions, the platform sentiment leaned clearly against Trump's action. This is a refreshing insight—showing that analyzing approval signals like upvotes can expose the dominant stance of a community.

Still, the analysis faced noise. The word cloud, for instance, is overly "cloudy," cluttered with filler terms—pronouns, generic verbs, and function words—distracting from actual sentiment-bearing keywords. The word count metric suffers similarly, with "Iran" and "war" being the only standout terms. This points to the need for a stronger data cleaning and preprocessing pipeline in future work—more aggressive stopword filtering, *lemmatization* (*reduction to dictionary form. Ex.: running* > *run*), and perhaps custom dictionaries.

A compelling next step would be to extract tuples combining sentiment-driven words with strong action verbs (e.g., "will," "can't," "hate," "trust") to uncover more contextually rich sentiment expressions. This could help reclassify some "neutral" comments that actually carry strong emotional cues. More broadly, combining score-weighted sentiment, advanced phrase extraction, and a topic modeling layer could lead to a sharper and more actionable understanding of how public opinion shapes around a political brand in times of crisis.