Project Report: Reddit Cryptocurrency Sentiment Analysis

AUTHOR
Wenbo Liu (wl2939), Lilly Loghmani (lql2103), Yixin Xiao, Thomas Bordino

PUBLISHED
February 19, 2025

1. Introduction

Project Overview

This project aims to analyze sentiment trends in Reddit cryptocurrency discussions. The goal is to collect, clean, preprocess, and analyze textual and numerical data from Reddit posts, focusing on Bitcoin-related discussions. The final analysis highlights patterns in engagement metrics (upvotes, comments) and explores their relationship with sentiment and time-based features.

The methodology is designed to be easily adaptable for any cryptocurrency, making it highly relevant for various applications, including trend prediction and even in-game trading strategies. By modifying the focus to different coins, the approach can be extended to analyze sentiment dynamics across multiple assets, offering valuable insights for traders and investors. The integration of sentiment analysis with engagement metrics allows for the development of predictive models that can inform trading strategies or game-based economic simulations. This adaptability ensures that the project remains a versatile tool for both market analysis and interactive financial applications.

Data Source

The data is collected from the **r/cryptocurrency** subreddit using the **Reddit API (praw)**. The dataset contains metadata, including: - **Post Title & Content** - **Upvotes & Comments** - **Post Creation Time** - **Post URL**

2. Data Acquisition

Methodology

We used the `praw` library to fetch Reddit posts related to Bitcoin using the `subreddit.search()` function. The search query was set to “Bitcoin” to ensure relevance.

Code Implementation

```
import praw
import pandas as pd

reddit = praw.Reddit(
    client_id="your_client_id",
    client_secret="your_client_secret",
    user_agent="CryptoSentimentApp"
)

def fetch_reddit_posts(coin, num_posts):
    subreddit = reddit.subreddit("cryptocurrency")
    posts_data = []
    for post in subreddit.search(coin, limit=num_posts):
        posts_data.append([
            post.id, post.title, post.selftext, post.score, post.num_comments, post.url, post.created_ut
        ])
    df = pd.DataFrame(posts_data, columns=["Post_ID", "Title", "Content", "Upvotes", "Comments", "URL",
    return df
```

Dataset Sample

Post_ID	Title	Upvotes	Comments	Timestamp
1gqafju	Bitcoin cycle analysis	3577	701	2024-01-01
1h6yoqp	Bitcoin hits 100K	19972	342	2024-01-02

3. Data Cleaning & Preprocessing

Cleaning Steps

- **Convert timestamps to datetime format**
- **Remove special characters & URLs** from text
- **Handle missing values (`content` field filled with "No content")**
- **Remove duplicate posts based on URLs**

```
        def clean_reddit_data(df):
            df = df.copy()
            df.columns = df.columns.str.lower().str.replace(" ", "_")
            df["timestamp"] = pd.to_datetime(df["timestamp"], unit="s")
            df.drop_duplicates(subset=["url"], inplace=True)
            df["content"].fillna("No content", inplace=True)
            return df
```

Preprocessing Steps

- **Feature scaling (Log Transform for Upvotes & Comments)**
- **Extract time-based features (Hour, Day, Weekend Flag)**
- **Categorical encoding for `content`**

```
import numpy as np
from sklearn.preprocessing import StandardScaler

def preprocess_data(df):
    df = df.copy()
    df['day_of_week'] = df['timestamp'].dt.dayofweek
    df['hour_of_day'] = df['timestamp'].dt.hour
    df['is_weekend'] = df['day_of_week'].apply(lambda x: 1 if x >= 5 else 0)
    df['log_upvotes'] = np.log1p(df['upvotes'])
    df['log_comments'] = np.log1p(df['comments'])
    scaler = StandardScaler()
    df[['upvotes', 'comments']] = scaler.fit_transform(df[['upvotes', 'comments']])
    return df
```

---

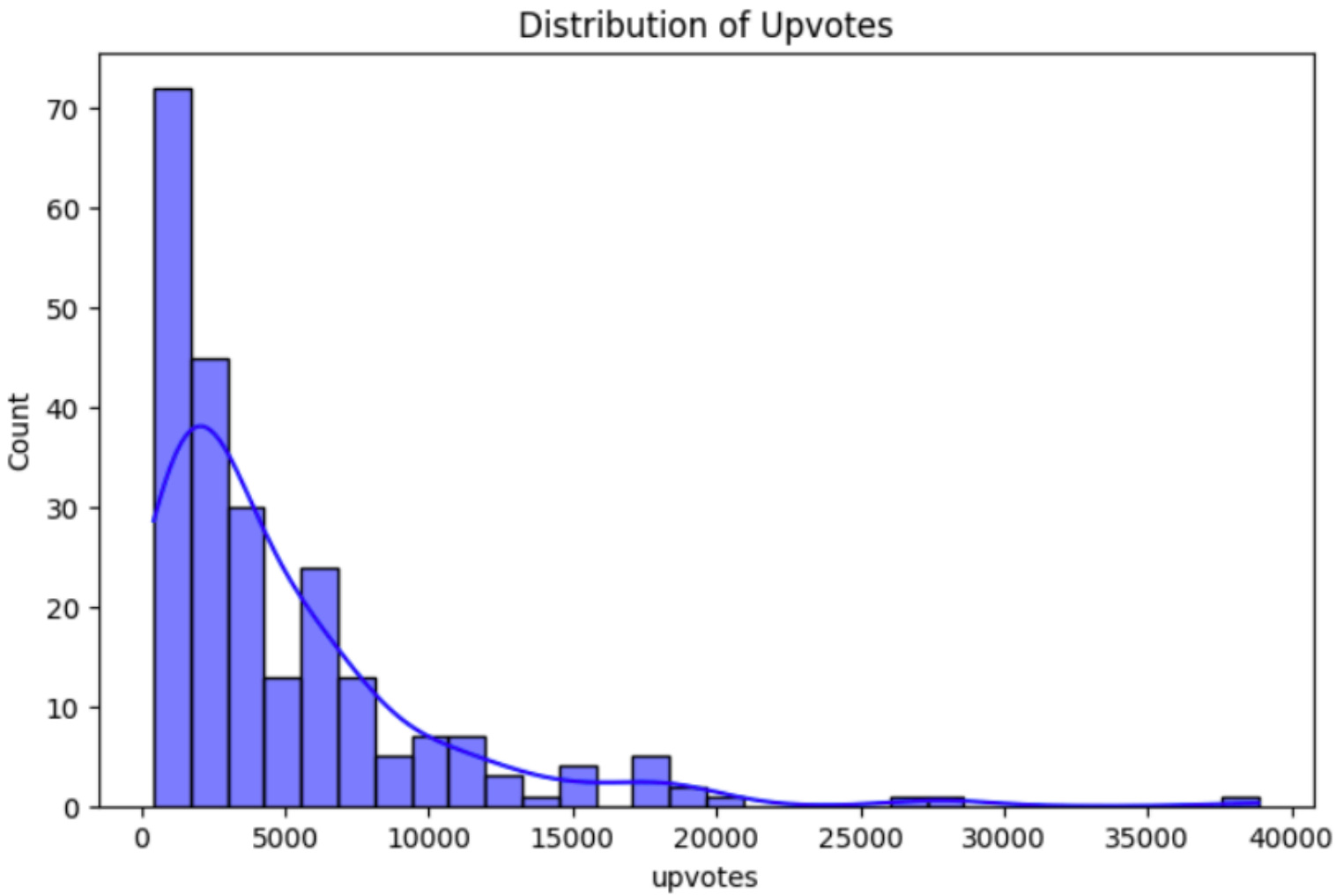
4. Exploratory Data Analysis (EDA)

Key Insights from Visualizations

- Upvotes and comments follow a right-skewed distribution, meaning most posts receive relatively low engagement, while a few highly popular posts accumulate significantly more interactions. This suggests that engagement is unevenly distributed, with viral content driving a large share of the activity.
- Engagement differs between weekdays and weekends, with upvotes tending to be higher on weekends. This could be due to users having more free time to browse and interact with posts, leading to increased visibility and participation in discussions.
- There is a positive correlation between upvotes and comments, indicating that posts receiving more upvotes also tend to generate more discussion. This suggests that highly engaging content not only attracts approval but also encourages further conversation within the community.

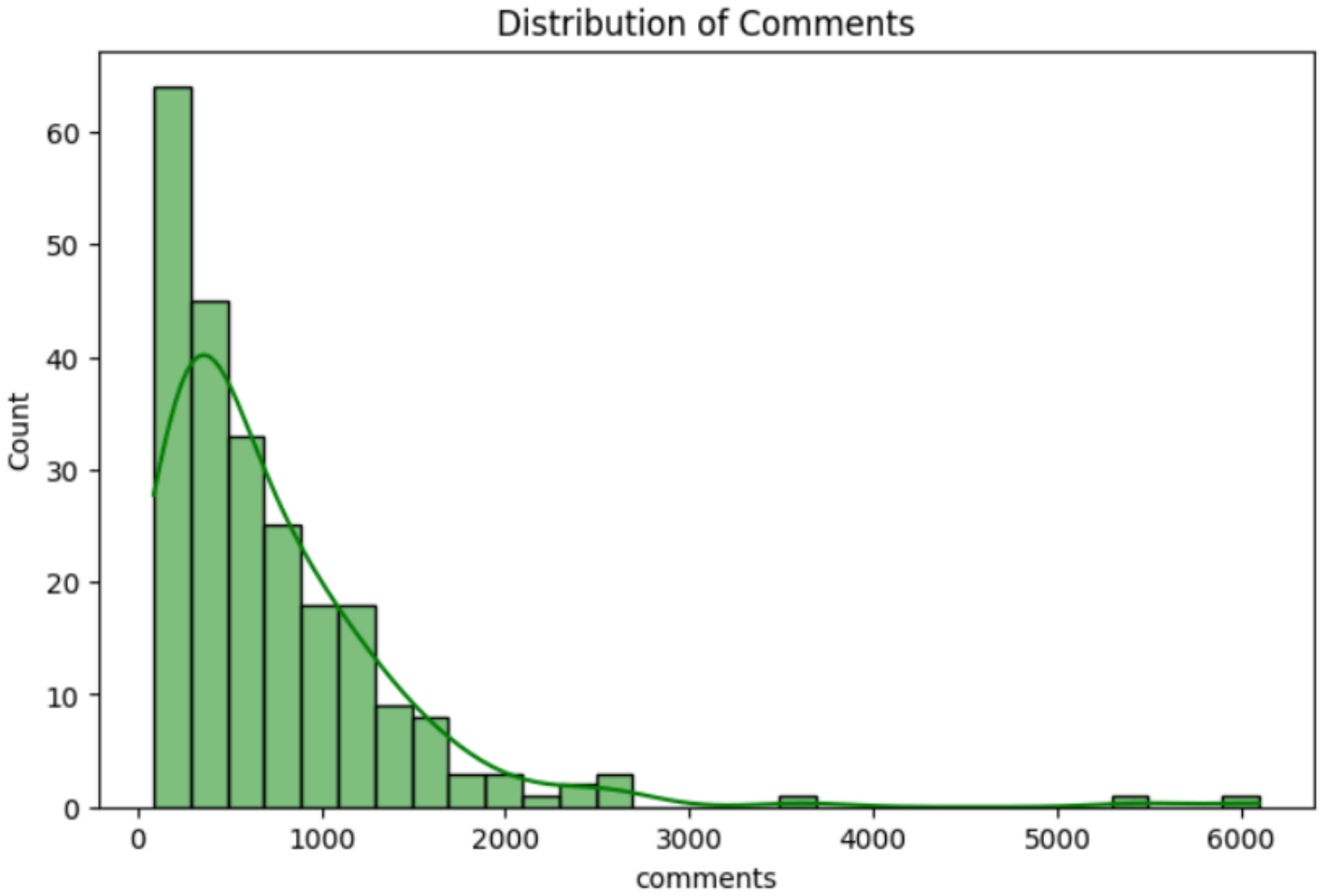
Visualization: Upvotes Distribution

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.histplot(df['upvotes'], bins=30, kde=True, color='blue')
plt.title("Distribution of Upvotes")
plt.show()
```



Visualization: Comments Distribution

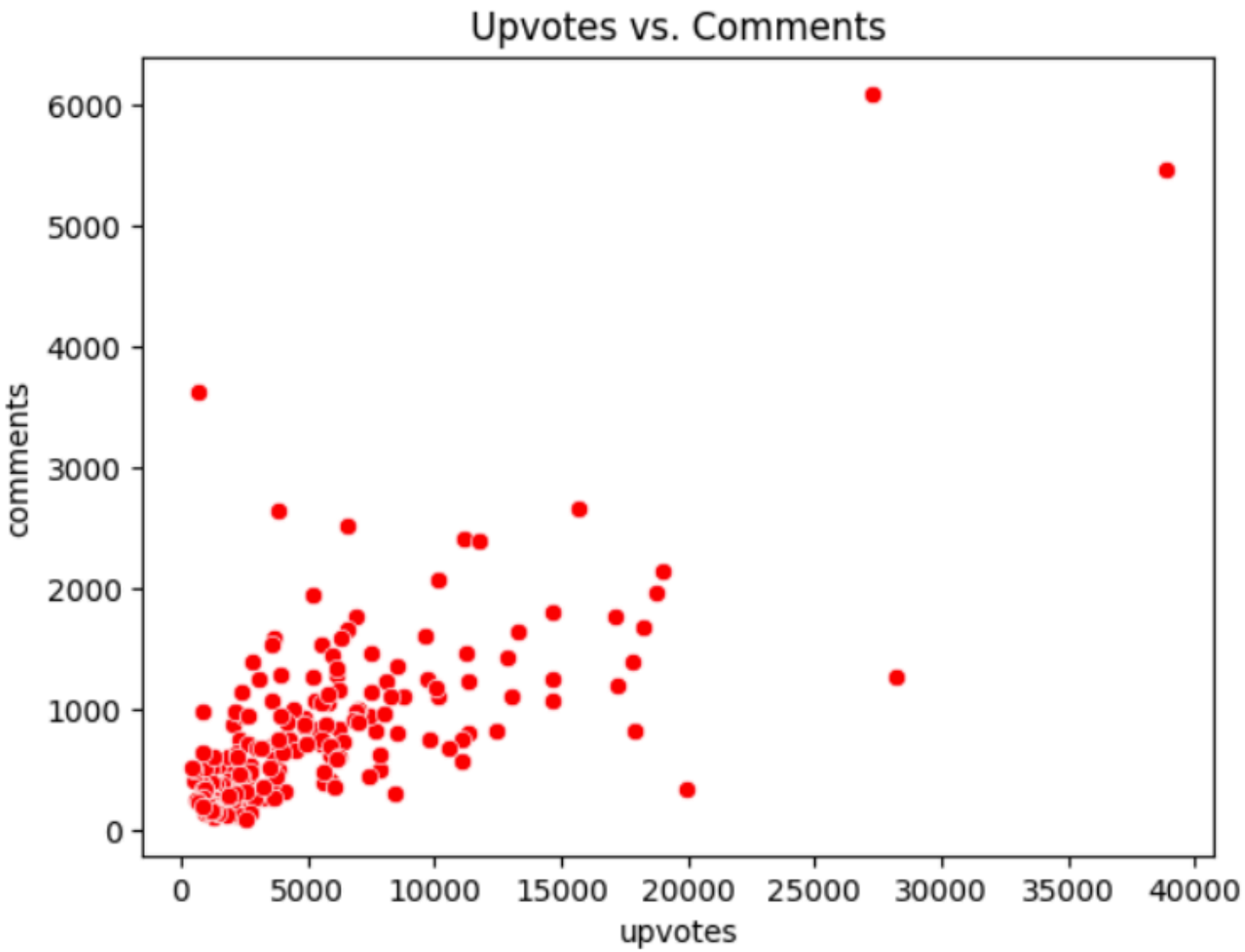
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.histplot(df['comments'], bins=30, kde=True, color='blue')
plt.title("Distribution of Comments")
plt.show()
```



Visualization: Comments And Upvotes

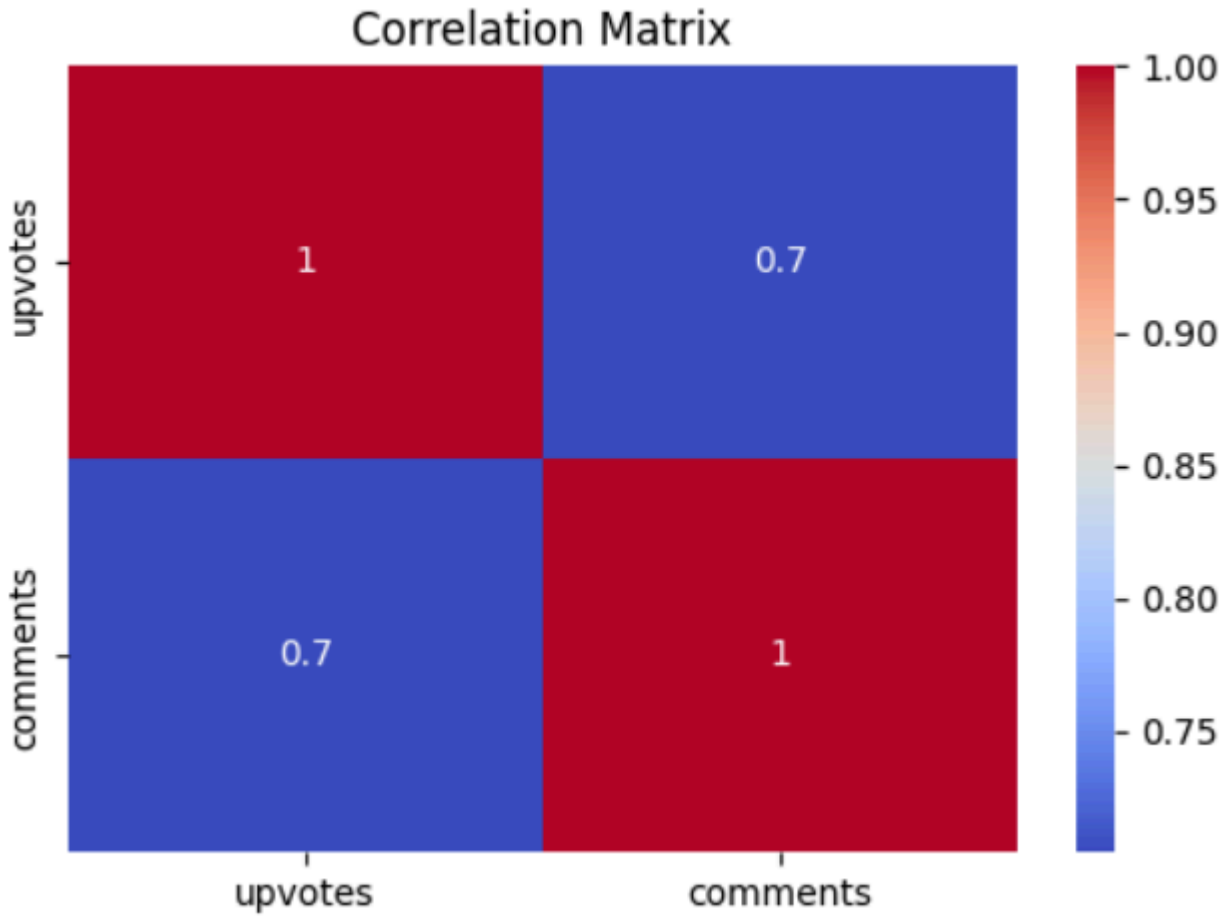
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.scatterplot(x='upvotes', y='comments', data=df, color='red')
```

```
plt.title("Upvotes vs. Comments")
plt.show()
```



Visualization: Correlation Heatmap

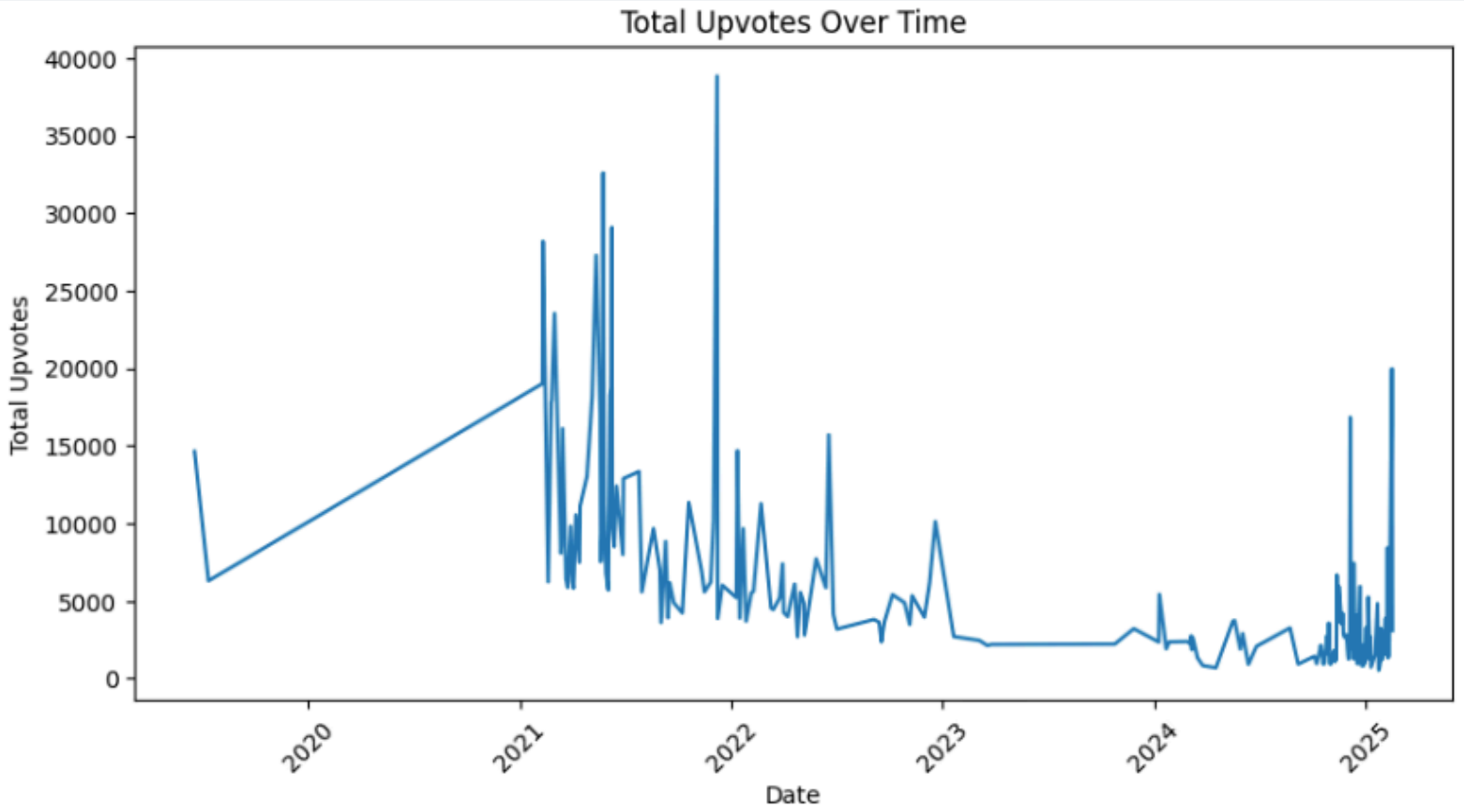
```
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```



Visualization: Upvotes over Time

```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df['timestamp'] = pd.to_datetime(df['timestamp'])
df['date'] = df['timestamp'].dt.date
df.groupby('date')['upvotes'].sum().plot(figsize=(10, 5))
plt.title("Total Upvotes Over Time")
plt.xlabel("Date")
plt.ylabel("Total Upvotes")
plt.xticks(rotation=45)
plt.show()
```

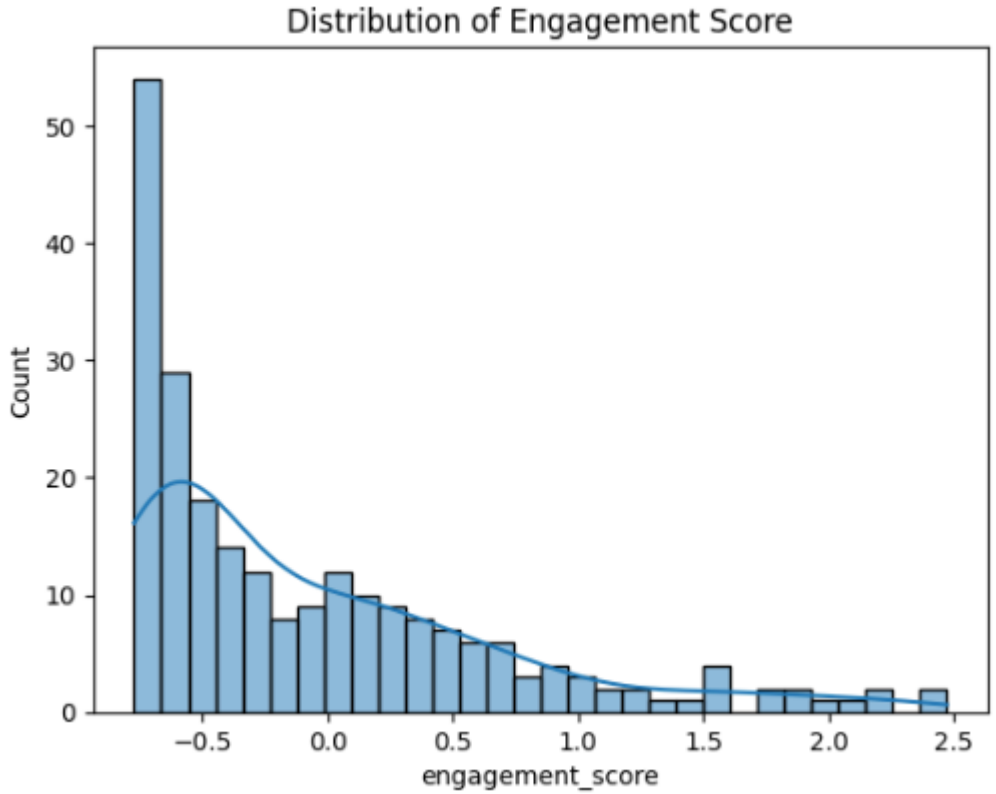


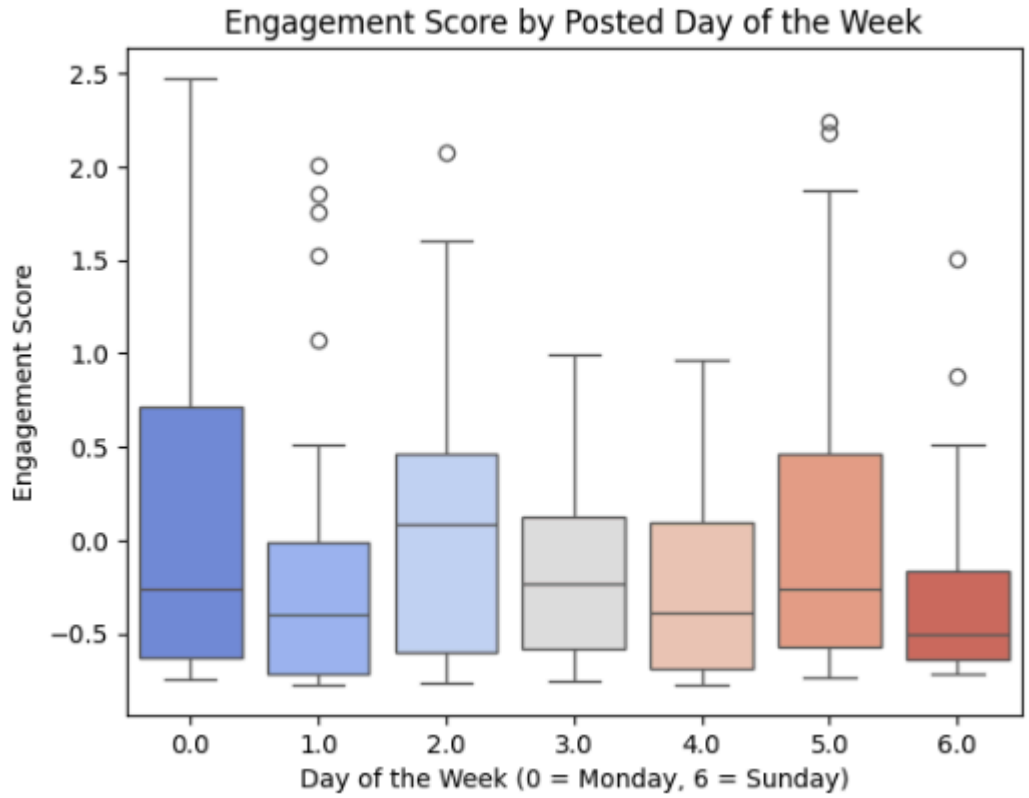
---

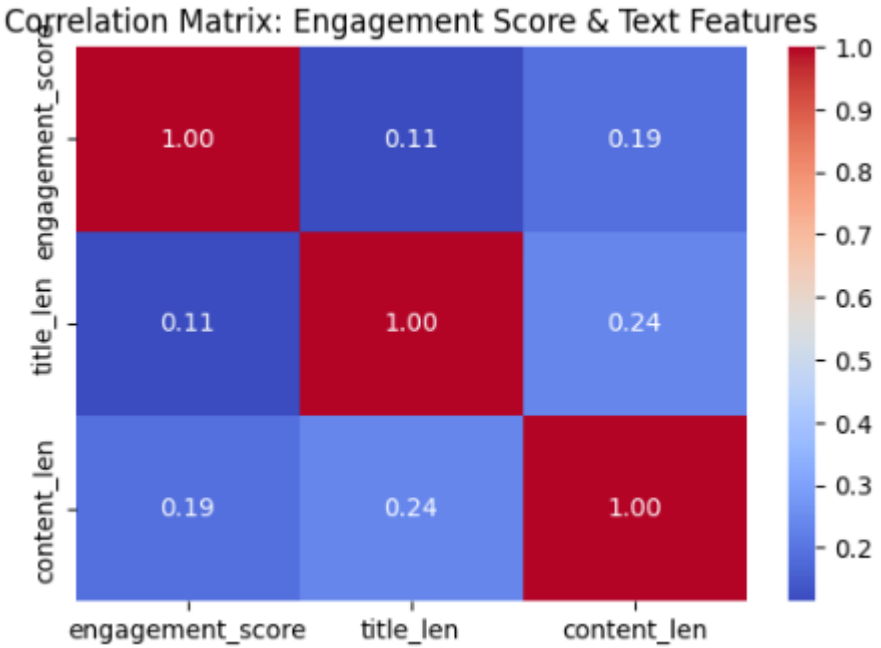
5. Feature Engineering

Added Features

Feature	Description
`log_upvotes`	Log transformation of upvotes because of right skewed distribution
`log_comments`	Log transformation of comments because of right skewed distribution
`title_len`	Number of words in post title
`content_len`	Number of words in post content
`is_working_hours`	Whether the post was made during business hours (9AM - 5PM)
`engagement_score`	Weighted score of upvotes & comments
`upvote_to_comment_ratio`	Ratio of upvotes to comments
`has_bitcoin`	Whether “Bitcoin” appears in the title


Distribution of Engagement Score
Count
engagement_score
0
10
20
30
40
50
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


Engagement Score by Posted Day of the Week
Engagement Score
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
0.0
1.0
2.0
3.0
4.0
5.0
6.0
Day of the Week (0 = Monday, 6 = Sunday)



6. Conclusion & Future Work

Findings

Sentiment polarity plays a crucial role in driving engagement on Reddit. Posts with a more positive sentiment tend to receive higher upvotes, suggesting that users are more likely to engage with optimistic or bullish discussions. This indicates that sentiment not only reflects market mood but also directly influences how widely content is shared and supported within the community.

Posts that include “Bitcoin” in the title tend to attract more upvotes, likely because Bitcoin remains the most recognized and influential cryptocurrency. This suggests that discussions explicitly mentioning Bitcoin are perceived as more relevant or valuable, drawing greater attention from the community. The effect may also be linked to algorithmic promotion, as posts with trending keywords like “Bitcoin” are more likely to be surfaced to users.

Engagement levels tend to be higher on weekends, possibly because users have more free time to browse and interact with cryptocurrency discussions. This pattern suggests that sentiment analysis and trading strategies could benefit from factoring in time-based trends, as increased activity during weekends may amplify sentiment-driven market reactions.

Next Steps

- NLP techniques like TF-IDF and LLM-based sentiment analysis help extract meaningful insights from Reddit discussions, identifying key terms and sentiment trends.
- Predictive models can be built to analyze the relationship between Reddit sentiment and Bitcoin price movements, potentially aiding in market forecasting.
- Fine-tuning LLMs for finance-specific sentiment classification can improve accuracy, making sentiment analysis more relevant for trading and investment strategies.

This report summarizes the entire process, from data collection to analysis. The next phase will explore predictive modeling using the cleaned dataset.