Project Report: Reddit Cryptocurrency Sentiment Analysis

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.

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_utc
```

```
        ])
    df = pd.DataFrame(posts_data, columns=["Post_ID", "Title", "Content", "Upvotes", "Comments", "URL", "Timestamp"])
    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 & comments have a right-skewed distribution.**
- **Weekday vs. Weekend engagement shows higher upvotes on weekends.**
- **Correlation between upvotes and comments is positive.**

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: Correlation Heatmap

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

```python
df['is_working_hours'] = df['hour_of_day'].apply(lambda x: 1 if 9 <= x <= 17 else 0)
df['title_len'] = df['title'].apply(lambda x: len(str(x).split()))
df['content_len'] = df['content'].apply(lambda x: len(str(x).split()))
df['engagement_score'] = df['upvotes'] * 0.7 + df['comments'] * 0.3
df['upvote_to_comment_ratio'] = df['upvotes'] / (df['comments'] + 1)
df['has_bitcoin'] = df['title'].apply(lambda x: 1 if 'bitcoin' in str(x).lower() else 0)
```

6. Conclusion & Future Work

Findings

- Sentiment polarity influences engagement (higher upvotes for positive sentiment).
- Posts with "Bitcoin" in the title tend to receive higher upvotes.
- Weekends show higher engagement.

Next Steps

- Apply NLP techniques (TF-IDF, LLM sentiment analysis).
- Build predictive models for Bitcoin price movement based on Reddit sentiment.
- Experiment with fine-tuned LLMs for finance-specific sentiment classification.

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