## Dataset- Tweets

```
#Dataset Link: https://www.kaggle.com/datasets/kaushiksuresh147/bitcoin-tweets
#Divide the large csv file into smaller chunks for easier storage due to its big size.
```

```
import pandas as pd
import os

file_urls = [
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-02-25.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-02-26.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-02-27.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-02-28.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-03-01.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-03-02.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-03-03.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-03-04.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-03-05.csv'
]

for url in file_urls:
    # Extract the file name
    file_name = url.split('/')[-1]

    # Read the CSV file
    df = pd.read_csv(url)

    # Save the DataFrame as CSV with a different file name
    file_name_without_extension = file_name.split('.')[0]
    new_file_name = f"{file_name_without_extension}.csv"
    df.to_csv(new_file_name, index=False)

    # Check the size of the saved file
    file_size_mb = os.path.getsize(new_file_name) / (1024 * 1024)  # in MB
    print(f"File {new_file_name} size: {file_size_mb:.2f} MB")

```

```
File bitcoinTweets_2023-02-25.csv size: 0.97 MB
File bitcoinTweets_2023-02-26.csv size: 9.32 MB
File bitcoinTweets_2023-02-27.csv size: 6.40 MB
File bitcoinTweets_2023-02-28.csv size: 10.86 MB
File bitcoinTweets_2023-03-01.csv size: 13.57 MB
File bitcoinTweets_2023-03-02.csv size: 9.28 MB
File bitcoinTweets_2023-03-03.csv size: 9.85 MB
File bitcoinTweets_2023-03-04.csv size: 9.72 MB
File bitcoinTweets_2023-03-05.csv size: 3.69 MB
```

```
#Concatenate the Tweets into one file for Analysis
```

```
import pandas as pd

file_urls = [
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-02-25.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-02-26.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-02-27.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-02-28.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-03-01.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-03-02.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-03-03.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-03-04.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/bitcoinTweets_2023-03-05.csv'
]

dfs = []

for url in file_urls:
    # Read the CSV file
    df = pd.read_csv(url)

    # Append the DataFrame to the list
    dfs.append(df)

```

```
# Combine all DataFrames into a single DataFrame
combined_df = pd.concat(dfs)

# Save the combined DataFrame as a CSV file
combined_df.to_csv('bitcoinTweets.csv', index=False)
```

```
bitcoinTweets= pd.read_csv("/content/bitcoinTweets.csv")
bitcoinTweets.head()
```

```
df = bitcoinTweets.copy()
```

```
df.head()
```

```
df.info()
```

```
print(df.columns)
```

```
import pandas as pd
import numpy as np
import datetime as dt

# Convert 'date' column to datetime64[ns]
df['date'] = pd.to_datetime(df['date'], format='%Y-%m-%d %H:%M:%S')

# Display the updated dataset
df.info()
```

```
# Check the size of the dataset
num_rows, num_cols = df.shape
print("Number of rows:", num_rows)
print("Number of columns:", num_cols)
```

```
import pandas as pd
import sys
# Calculate the size of the dataset in megabytes
dataset_size_mb = sys.getsizeof(df) / (1024 * 1024)
print("Dataset size:", round(dataset_size_mb, 2), "MB")
```

```
column_names = df.columns
print(column_names)
```

```
# Change the data type of date
date_dtype = df['date'].dtypes
print(date_dtype)
```

```
# Convert the 'date' column to a date format with errors set to 'coerce'
df['date'] = pd.to_datetime(df['date'], errors='coerce')

# Save the modified dataset
df.to_csv('/content/bitcoinTweets_updated.csv', index=False)
```

```
df = pd.read_csv('/content/bitcoinTweets_updated.csv')
df['date'] = pd.to_datetime(df['date'], errors='coerce')
date_dtype = df['date'].dtypes
print(date_dtype)
```

```
column_names = df.columns
print(column_names)
```

```
df.head()
```

# Bitcoin price

```
#Cryptocompare Website
#Dataset Link # https://min-api.cryptocompare.com/data/v2/histohour
```

```
import requests
import pandas as pd

# API endpoint for CryptoCompare
url = "https://min-api.cryptocompare.com/data/v2/histohour"

# Parameters for API request
coin_symbol = "BTC"
currency = "USD"
timestamp_start = 1677110400  # February 22, 2023, 00:00:00 in Unix timestamp
timestamp_end = 1678022399    # March 10, 2023, 23:59:59 in Unix timestamp

# API request payload
payload = {
    "fsym": coin_symbol,
    "tsym": currency,
    "toTs": timestamp_end,
    "limit": 336,  # Number of hours between the two timestamps
    "aggregate": 1,
    "e": "CCCAGG"  # CryptoCompare aggregate index
}

# Send API request
response = requests.get(url, params=payload)

# Check if the request was successful
if response.status_code == 200:
    # Process the response data
    data = response.json()
    # The 'Data' field contains the hourly Bitcoin prices
    bitcoin_prices = data['Data']['Data']

    # Create a DataFrame from the Bitcoin price data
    df = pd.DataFrame(bitcoin_prices, columns=["time", "close", "high", "low", "open", "volumefrom", "volumeto"])
    df["time"] = pd.to_datetime(df["time"], unit="s")  # Convert Unix timestamp to datetime

    # Save DataFrame to CSV file
    df.to_csv("bitcoin_prices_2023-02-22_to_2023-03-10.csv", index=False)
    print("Bitcoin data from February 22 to March 10, 2023, saved to 'bitcoin_prices_2023-02-22_to_2023-03-10.csv' file.")
else:
    print("Failed to fetch Bitcoin data.")

```

```
Bitcoin data from February 22 to March 10, 2023, saved to 'bitcoin_prices_2023-02-22_to_2023-03-10.csv' file.
```

```
df_bitcoin = pd.read_csv("/content/bitcoin_prices_2023-02-22_to_2023-03-10.csv")
```

```
df_price = df_bitcoin.copy()
df_price['time'] = pd.to_datetime(df_price['time'])

df_price['Date'] = df_price['time'].dt.date
df_price['Time'] = df_price['time'].dt.time

df_price.head(2)

```

| | time | close | high | low | open | volumefrom | volumeto | Date | Time |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 2023-02-19 13:00:00 | 24682.03 | 24715.82 | 24682.03 | 24707.39 | 903.97 | 22335943.28 | 2023-02-19 | 13:00:00 |

```

df_price['Date'] = pd.to_datetime(df_price['Date'], format='%Y-%m-%d').dt.date

```

```
df_price.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 337 entries, 0 to 336
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   time        337 non-null    datetime64[ns]
 1   close       337 non-null    float64
 2   high        337 non-null    float64
 3   low         337 non-null    float64
 4   open        337 non-null    float64
 5   volumefrom  337 non-null    float64
 6   volumeto    337 non-null    float64
 7   Date        337 non-null    object
 8   Time        337 non-null    object
dtypes: datetime64[ns](1), float64(6), object(2)
memory usage: 23.8+ KB
```

```
crypto_usd = df_price.copy()
# Drop duplicates Currency
print('bitcoin shape before droping duplicates', crypto_usd.shape)
duplicates_removed = crypto_usd.shape[0]
crypto_usd = crypto_usd.drop_duplicates(subset=['time'])
print('bitcoin shape after droping duplicates', crypto_usd.shape)
duplicates_removed -= crypto_usd.shape[0]
print('duplicates removed', duplicates_removed)
crypto_usd.head(2)
```

```
bitcoin shape before droping duplicates (337, 9)
bitcoin shape after droping duplicates (337, 9)
duplicates removed 0
```

| | time | close | high | low | open | volumefrom | volumeto | Date | Time |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 2023-02-19 13:00:00 | 24682.03 | 24715.82 | 24682.03 | 24707.39 | 903.97 | 22335943.28 | 2023-02-19 | 13:00:00 |

```
# Create the new 'volume' column
crypto_usd['volume'] = crypto_usd['volumeto'] - crypto_usd['volumefrom']

# Display the updated DataFrame
crypto_usd.head(2)
```

| | time | close | high | low | open | volumefrom | volumeto | Date | Time | |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2023-02-19 13:00:00 | 24682.03 | 24715.82 | 24682.03 | 24707.39 | 903.97 | 22335943.28 | 2023-02-19 | 13:00:00 | 22 |

```
# Sort the dataframe by the 'time' column
crypto_usd = crypto_usd.sort_values('time')

# Calculate market cap
crypto_usd['marketcap'] = crypto_usd['close'] * crypto_usd['volumeto']

# Calculate price difference delta
crypto_usd['price_delta'] = crypto_usd['close'].diff()

# Display the updated dataframe
print(crypto_usd.head())
```

```
                 time     close      high       low      open  volumefrom  \
0 2023-02-19 13:00:00  24682.03  24715.82  24682.03  24707.39      903.97
1 2023-02-19 14:00:00  24765.79  24792.85  24679.21  24682.03     1220.29
2 2023-02-19 15:00:00  24928.21  25022.49  24751.96  24765.79     5074.50
3 2023-02-19 16:00:00  24786.44  25175.28  24704.53  24928.21     7094.72
4 2023-02-19 17:00:00  24364.95  24806.64  24346.17  24786.44     6896.84

       volumeto        Date      Time        volume     marketcap  price_delta
0  2.233594e+07  2023-02-19  13:00:00  2.233504e+07  5.512964e+11          NaN
1  3.020300e+07  2023-02-19  14:00:00  3.020178e+07  7.480012e+11        83.76
2  1.263085e+08  2023-02-19  15:00:00  1.263034e+08  3.148644e+12       162.42
3  1.770671e+08  2023-02-19  16:00:00  1.770600e+08  4.388863e+12      -141.77
4  1.693379e+08  2023-02-19  17:00:00  1.693310e+08  4.125910e+12      -421.49
```

```
#Saved the Data with all additional attributes so it is easier to extract for analysis
```

```
import pandas as pd

# Save the DataFrame as CSV
crypto_usd.to_csv('BitcoinPrice.csv', index=False)

# Print the file name
print("Data saved as BitcoinPricePreprocessed.csv")

```

```
Data saved as BitcoinPricePreprocessed.csv
```

## Tweets Cleaning

```
import pandas as pd

# Read the CSV file
tweets_raw_file = pd.read_csv('/content/bitcoinTweets_updated.csv')

# Convert 'date' column to datetime
tweets_raw_file['date'] = pd.to_datetime(tweets_raw_file['date'], errors='coerce')

# Check the data type of 'date' column
date_dtype = tweets_raw_file['date'].dtypes

# Print the data type
print(date_dtype)

```

```
datetime64[ns]
```

```
tweets_raw_file.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169761 entries, 0 to 169760
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype         
---  ------            --------------   -----         
 0   user_name         169751 non-null  object        
 1   user_location     84090 non-null   object        
 2   user_description  158666 non-null  object        
 3   user_created      169761 non-null  object        
 4   user_followers    169761 non-null  float64       
 5   user_friends      169761 non-null  float64       
 6   user_favourites   169761 non-null  float64       
 7   user_verified     169761 non-null  bool          
 8   date              169761 non-null  datetime64[ns]
 9   text              169761 non-null  object        
 10  hashtags          168954 non-null  object        
 11  source            168954 non-null  object        
 12  is_retweet        168954 non-null  float64       
dtypes: bool(1), datetime64[ns](1), float64(4), object(7)
memory usage: 15.7+ MB
```

```
import pandas as pd
import re
import string
from tqdm import tqdm
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import nltk
nltk.download('stopwords')


tweets_raw_file = "/content/bitcoinTweets_updated.csv"
tweets_clean_file = "bitcoinTweets_cleaned.csv"

# Load the tweets DataFrame
tweets_df = pd.read_csv(tweets_raw_file)

# Preprocess the 'text' column
for i, text in tqdm(enumerate(tweets_df['text']), total=len(tweets_df['text'])):
```

```
    # Remove hashtags
    text = text.replace("#", "")

    # Remove URLs
    text = re.sub('https?://(?:[-\w.]|(?:%[\da-fA-F]{2}))+', '', text)

    # Remove mentions
    text = re.sub('@\\w+ *', '', text)

    # Convert to lowercase
    text = text.lower()

    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))

    # Tokenize the text
    tokens = text.split()

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [word for word in tokens if word not in stop_words]

    # Apply stemming
    stemmer = PorterStemmer()
    tokens = [stemmer.stem(word) for word in tokens]

    # Join the tokens back into a single string
    preprocessed_text = ' '.join(tokens)

    # Update the 'text' column with the preprocessed text
    tweets_df.loc[i, 'text'] = preprocessed_text

# Save the cleaned data to a new CSV file
tweets_df.to_csv(tweets_clean_file, header=True, encoding='utf-8', index=False)

print("Cleaned tweets data saved to 'bitcoinTweets_cleaned.csv'.")

```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
100%|██████████| 169761/169761 [11:21<00:00, 249.02it/s]
Cleaned tweets data saved to 'bitcoinTweets_cleaned.csv'.
```

```
import pandas as pd

tweets_clean_file = "bitcoinTweets_cleaned.csv"

# Load the cleaned tweets DataFrame
tweets_df = pd.read_csv(tweets_clean_file)
print('tweet shape before droping duplicates', tweets_df.shape)
# Remove duplicate rows based on all columns
tweets_df.drop_duplicates(inplace=True)

# Save the deduplicated data to a new CSV file
tweets_df.to_csv(tweets_clean_file, header=True, encoding='utf-8', index=False)

print("Duplicates removed from the cleaned tweets data.")
print('tweet shape after droping duplicates', tweets_df.shape)

```

```
tweet shape before droping duplicates (169761, 13)
Duplicates removed from the cleaned tweets data.
tweet shape after droping duplicates (168685, 13)
```

```
import sys

# Get the size of the DataFrame in bytes
size_bytes = sys.getsizeof(tweets_df)

# Convert the size to megabytes
size_mb = size_bytes / (1024 * 1024)

# Print the size in MB
print(f"Size of tweets_df: {size_mb} MB")

```

```
Size of tweets_df: 137.04691982269287 MB
```

```
1 print(tweets_df.shape)
```

```
(168685, 13)
```

```
1 tweets_df.head()
```

| | user_name | user_location | user_description | user_created | user_followers | user_friends | u |
|---|---|---|---|---|---|---|---|
| **0** | Irk | Vancouver, WA | Irk started investing in the stock market in 1... | 2018-08-11 03:17:00 | 116.0 | 8.0 | |
| **1** | Xiang Zhang | NaN | Professional Software Engineer ð□□»ð□□□Crypto ... | 2011-01-11 01:37:00 | 42.0 | 22.0 | |
| **2** | Rhizoo | NaN | researcher. local maxima dunningâ□□kruger spec... | 2019-04-03 18:09:00 | 778.0 | 627.0 | |
| **3** | Hari Marquez | Las Vegas, NV | Donâ□□t trust, verify. #Bitcoin \| El Salvador ... | 2014-01-17 23:04:00 | 222.0 | 521.0 | |
| **4** | Bitcoin Candle Bot | Brazil | Robot that posts the closure of the bitcoin da... | 2021-01-06 01:36:00 | 40.0 | 4.0 | |

```
1 tweets_df.min(axis=0)
```

```
<ipython-input-33-14ceec5556a9>:1: FutureWarning: The default value of numeric_only in DataFrame.min is deprecated. In a future version
  tweets_df.min(axis=0)
user_created       2006-09-04 17:48:00
user_followers                     0.0
user_friends                       0.0
user_favourites                    0.0
user_verified                    False
date               2023-02-25 20:49:00
is_retweet                         0.0
dtype: object
```

```
1 tweets_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 168685 entries, 0 to 169760
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   user_name         168675 non-null  object 
 1   user_location     83730 non-null   object 
 2   user_description  158165 non-null  object 
 3   user_created      168685 non-null  object 
 4   user_followers    168685 non-null  float64
 5   user_friends      168685 non-null  float64
 6   user_favourites   168685 non-null  float64
 7   user_verified     168685 non-null  bool   
 8   date              168685 non-null  object 
 9   text              168673 non-null  object 
 10  hashtags          167953 non-null  object 
 11  source            167953 non-null  object 
 12  is_retweet        167953 non-null  float64
dtypes: bool(1), float64(4), object(8)
memory usage: 16.9+ MB
```

```
1 #Filter Tweets and hashtag for Bitcoin only
```

```
import pandas as pd
import numpy as np

# Filter tweets containing specific keywords or hashtags related to Bitcoin
bitcoin_keywords = ['bitcoin', 'btc', 'crypto']
bitcoin_hashtags = ['#bitcoin', '#btc', '#cryptocurrency']

# Drop rows with missing values in the 'text' column
tweets_df.dropna(subset=['text'], inplace=True)

# Apply the filtering condition
filtered_tweets_df = tweets_df[tweets_df['text'].str.contains('|'.join(bitcoin_keywords + bitcoin_hashtags), case=False)]

# Print the filtered tweets DataFrame
print(filtered_tweets_df)

```

```
                              user_name  user_location  \
0                                  Irk  Vancouver, WA
1                          Xiang Zhang            NaN
2                              Rhizoo             NaN
3                         Hari Marquez  Las Vegas, NV
4                   Bitcoin Candle Bot         Brazil
...                                ...            ...
169756  L3X - ð□□°ð□□»ð□□´ðð□□ð□□□ v5.3            NaN
169757                       FunFacts.AI            NaN
169758               Word On Cryptoð□□□    SocialMedia
169759           BTC Status Alert ð□□□          Japan
169760                      Jon Padilha            NaN

                                       user_description  \
0       Irk started investing in the stock market in 1...
1       Professional Software Engineer ð□□»ð□□□Crypto ...
2       researcher. local maxima dunningâ□□kruger spec...
3       Donâ□□t trust, verify. #Bitcoin | El Salvador ...
4       Robot that posts the closure of the bitcoin da...
...                                                   ...
169756  #Bitcoin & #Crypto #Trading #Strategy\nFollow ...
169757  I post a Fun Fact every minute\n\n#OpenAI #Cha...
169758                                  ð□□¯UNDERGROUND
169759  Tweet data that will help you consider #BTC pr...
169760    Daytrade Institucional\nForex - BTC - BinÃ¡rias

              user_created  user_followers  user_friends  user_favourites  \
0       2018-08-11 03:17:00           116.0           8.0           4580.0
1       2011-01-11 01:37:00            42.0          22.0              5.0
2       2019-04-03 18:09:00           778.0         627.0          32005.0
3       2014-01-17 23:04:00           222.0         521.0          13052.0
4       2021-01-06 01:36:00            40.0           4.0              1.0
...                     ...             ...           ...              ...
169756  2022-04-05 22:51:00           182.0          14.0           2978.0
169757  2022-05-24 12:26:00            24.0          18.0              1.0
169758  2021-06-18 16:30:00            41.0          15.0            316.0
169759  2019-07-21 11:28:00         45666.0           2.0             15.0
169760  2019-09-22 13:54:00           101.0           3.0             35.0

        user_verified                date  \
0               False  2023-02-25 23:59:00
1               False  2023-02-25 23:59:00
2               False  2023-02-25 23:59:00
3               False  2023-02-25 23:59:00
4               False  2023-02-25 23:59:00
...               ...                  ...
169756          False  2023-03-05 18:52:00
169757          False  2023-03-05 18:52:00
169758          False  2023-03-05 18:52:00
169759          False  2023-03-05 18:51:00
169760          False  2023-03-05 18:51:00

                                                   text  \
0       bitcoin btc rest crypto ye bitcoin cryptocurr ...
1       retriev invest fund current ongo tidexcoin kic...
2       bull save monthli thread today good shit bitco...
3              el salvador shape futur bitcoin membvk32cn
4       candl day 25022023 close open 2319406 high 232...
```

```
import sys

# Get the size of the DataFrame in bytes
size_bytes = sys.getsizeof(filtered_tweets_df)

# Convert the size to megabytes
```

```
size_mb = size_bytes / (1024 * 1024)

# Print the size in MB
print(f"Size of tweets_df: {size_mb} MB")

```

```
Size of tweets_df: 136.2565097808838 MB
```

## Sentiment analysis with Vader

```
!!pip install vaderSentiment

```

```
['Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/',
 'Collecting vaderSentiment',
 '  Downloading vaderSentiment-3.3.2-py2.py3-none-any.whl (125 kB)',
 '\x1b[?25l     \x1b[90m━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━\x1b[0m \x1b[32m0.0/126.0 kB\x1b[0m \x1b[31m?\x1b[0m eta \x1b[36m-:--:--\x1b[0m',
 '\x1b[2K     \x1b[90m━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━\x1b[0m \x1b[32m126.0/126.0 kB\x1b[0m \x1b[31m3.5 MB/s\x1b[0m eta \x1b[36m0:00:00\x1b[0m',
 '\x1b[?25hRequirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from vaderSentiment) (2.27.1)',
 'Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (1.26.16)',
 'Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (2023.5.7)',
 'Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (2.0.12)',
 'Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (3.4)',
 'Installing collected packages: vaderSentiment',
 'Successfully installed vaderSentiment-3.3.2']
```

```
import pandas as pd
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from tqdm import tqdm

# Assuming you have already cleaned the tweets and stored them in the 'text' column of `df_clean`
df_clean = filtered_tweets_df.copy()

# Handle NaN values in the 'text' column
df_clean['text'].fillna('', inplace=True)

# Initialize the SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()

# Perform sentiment analysis and store the compound scores
compound = []
for i, s in tqdm(enumerate(df_clean['text']), total=len(df_clean['text'])):
    if isinstance(s, str):  # Check if the tweet is a string
        vs = analyzer.polarity_scores(s)
        compound.append(vs["compound"])
    else:
        compound.append(0.0)  # Assign a neutral score for non-string tweets

# Add the compound scores to the dataframe
df_clean["compound"] = compound

# Save the updated dataframe to a new CSV file
df_clean.to_csv("bitcoinTweets_sentiment.csv", index=False)

print("Sentiment analysis completed and data saved to 'bitcoinTweets_sentiment.csv'.")

```

```
100%|██████████| 167655/167655 [00:29<00:00, 5594.51it/s]
Sentiment analysis completed and data saved to 'bitcoinTweets_sentiment.csv'.
```

```
df_clean.head()
```

| | user_name | user_location | user_description | user_created | user_followers | user_friends | us |
|---|---|---|---|---|---|---|---|
| **0** | Irk | Vancouver, WA | Irk started investing in the stock market in 1... | 2018-08-11 03:17:00 | 116.0 | 8.0 | |
| **1** | Xiang Zhang | NaN | Professional Software Engineer ð□□»ð□□□Crypto ... | 2011-01-11 01:37:00 | 42.0 | 22.0 | |
| **2** | Rhizoo | NaN | researcher. local maxima dunningâ□□kruger spec... | 2019-04-03 18:09:00 | 778.0 | 627.0 | |
| **3** | Hari Marquez | Las Vegas, NV | Donâ□□t trust, verify. #Bitcoin \| El Salvador ... | 2014-01-17 23:04:00 | 222.0 | 521.0 | |

```
#Calculate a score for each tweet
```

```
# Perform sentiment analysis and calculate the scores
scores = []
for i, s in tqdm(df_clean.iterrows(), total=df_clean.shape[0]):
    vs = analyzer.polarity_scores(s["text"])
    score = vs["compound"] * (s["user_followers"] + 1) * (s["user_favourites"] + 1)
    scores.append(score)

# Add the scores to the dataframe
df_clean["score"] = scores

# Display the first two rows of the updated dataframe
print(df_clean.head(2))
```

```
100%|██████████| 167655/167655 [00:47<00:00, 3543.34it/s]     user_name  user_location  \
0          Irk  Vancouver, WA
1  Xiang Zhang            NaN

                                  user_description         user_created  \
0  Irk started investing in the stock market in 1...  2018-08-11 03:17:00
1  Professional Software Engineer ð□□»ð□□□Crypto ...  2011-01-11 01:37:00

   user_followers  user_friends  user_favourites  user_verified  \
0           116.0           8.0           4580.0          False
1            42.0          22.0              5.0          False

                 date                                               text  \
0  2023-02-25 23:59:00  bitcoin btc rest crypto ye bitcoin cryptocurr ...
1  2023-02-25 23:59:00  retriev invest fund current ongo tidexcoin kic...

                                      hashtags              source  \
0           ['Bitcoin', 'crypto', 'NeedsMoreCrash']     Twitter Web App
1  ['Tidexcoin', 'Kicurrency', 'LMY', 'GMK', 'SYR...  Twitter for iPhone

   is_retweet  compound        score
0         0.0   -0.4019 -215409.1563
1         0.0    0.0000       0.0000
```

```
df_clean.head(2)
```

| | user_name | user_location | user_description | user_created | user_followers | user_friends | us |
|---|---|---|---|---|---|---|---|
| **0** | Irk | Vancouver, WA | Irk started investing in the stock market in 1... | 2018-08-11 03:17:00 | 116.0 | 8.0 | |

```
import sys

# Get the size of the DataFrame in bytes
size_bytes = sys.getsizeof(df_clean)

# Convert the size to megabytes
size_mb = size_bytes / (1024 * 1024)

# Print the size in MB
print(f"Size of tweets_df: {size_mb} MB")

```

```
Size of tweets_df: 138.81472206115723 MB
```

```
# Check for unique values in the 'user_name' column
unique_user_names = df_clean['user_name'].unique()

# Check if all user names are unique
if len(unique_user_names) == len(df_clean):
    print("All user names are unique.")
else:
    print("There are duplicate user names.")
# Check the number of unique values in the 'user_name' column
num_unique_user_names = df_clean['user_name'].nunique()
print(f"The number of unique user names is: {num_unique_user_names}")
```

```
There are duplicate user names.
The number of unique user names is: 36054
```

```
#Split dataframe and save it into multiple files
```

```
date_column_format = df_clean['date'].dtypes

print(f"The format of the 'date' column is: {date_column_format}")
```

```
The format of the 'date' column is: object
```

```
#The purpose of dividing the DataFrame into smaller chunks and then merging them back together is to handle large datasets efficiently.
```

```

import pandas as pd
from datetime import datetime

n = 20000  # chunk row size
chunks_df = [df_clean[i:i+n] for i in range(0, df_clean.shape[0], n)]

sep_char = '~'
path = '/content'  # Specify the path where you want to save the CSV files

# Concatenate all the chunked DataFrames into a single DataFrame
concatenated_df = pd.concat(chunks_df)

# Get the minimum and maximum dates from the concatenated DataFrame
date_from = datetime.strptime(concatenated_df['date'].min(), '%Y-%m-%d %H:%M:%S').strftime('%Y-%m-%d %H:%M:%S')
date_to = datetime.strptime(concatenated_df['date'].max(), '%Y-%m-%d %H:%M:%S').strftime('%Y-%m-%d %H:%M:%S')

# Write the concatenated DataFrame into CSV
concatenated_df.to_csv(f"{path}/Chunk_{date_from}{sep_char}{date_to}.csv", header=True, index=False)
```

```
import sys

# Get the size of the DataFrame in bytes
size_bytes = sys.getsizeof(concatenated_df)

# Convert the size to megabytes
size_mb = size_bytes / (1024 * 1024)
```

```
 8
 9 # Print the size in MB
10 print(f"Size of tweets_df: {size_mb} MB")
11
```

```
Size of tweets_df: 138.81472206115723 MB
```

```
1 concatenated_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 167655 entries, 0 to 169760
Data columns (total 15 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   user_name         167645 non-null  object
 1   user_location     83351 non-null   object
 2   user_description  157326 non-null  object
 3   user_created      167655 non-null  object
 4   user_followers    167655 non-null  float64
 5   user_friends      167655 non-null  float64
 6   user_favourites   167655 non-null  float64
 7   user_verified     167655 non-null  bool
 8   date              167655 non-null  object
 9   text              167655 non-null  object
 10  hashtags          167145 non-null  object
 11  source            167145 non-null  object
 12  is_retweet        167145 non-null  float64
 13  compound          167655 non-null  float64
 14  score             167655 non-null  float64
dtypes: bool(1), float64(6), object(8)
memory usage: 19.3+ MB
```

```
1 concatenated_df.head()
```

| | user_name | user_location | user_description | user_created | user_followers | user_friends | us |
|---|---|---|---|---|---|---|---|
| **0** | Irk | Vancouver, WA | Irk started investing in the stock market in 1... | 2018-08-11 03:17:00 | 116.0 | 8.0 | |
| **1** | Xiang Zhang | NaN | Professional Software Engineer ð□□»ð□□□Crypto ... | 2011-01-11 01:37:00 | 42.0 | 22.0 | |
| **2** | Rhizoo | NaN | researcher. local maxima dunningâ□□kruger spec... | 2019-04-03 18:09:00 | 778.0 | 627.0 | |
| **3** | Hari Marquez | Las Vegas, NV | Donâ□□t trust, verify. #Bitcoin \| El Salvador ... | 2014-01-17 23:04:00 | 222.0 | 521.0 | |
| **4** | Bitcoin Candle Bot | Brazil | Robot that posts the closure of the bitcoin da... | 2021-01-06 01:36:00 | 40.0 | 4.0 | |

```
1 #Sentiment Level
```

```
1 #The compound score represents the overall sentiment intensity of the text, taking into account the positive, negative, and neutral scores
```

```
1 import nltk
2 nltk.download('vader_lexicon')
3 from nltk.sentiment import SentimentIntensityAnalyzer
4 import pandas as pd
5 import numpy as np
6
7 # Initialize VADER sentiment analyzer
```

```
sia = SentimentIntensityAnalyzer()

# Calculate compound scores
concatenated_df['compound'] = concatenated_df['text'].apply(lambda x: sia.polarity_scores(x)['compound'])

# Find the range of compound values
compound_min = concatenated_df['compound'].min()
compound_max = concatenated_df['compound'].max()

# Define custom bin edges based on quantiles
bin_edges = np.linspace(compound_min, compound_max, num=6)  # Adjust the 'num' parameter as needed

# Define labels
labels = ['Extreme Negative', 'Negative', 'Neutral', 'Positive', 'Extreme Positive']

# Assign sentiment levels based on custom bins
concatenated_df['sentiment_level'] = pd.cut(concatenated_df['compound'], bins=bin_edges, labels=labels, include_lowest=True)

# Save the updated dataframe as a new CSV file
concatenated_df.to_csv('updated_sentiment_data.csv', index=False)

```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
```

```
concatenated_df_2 = pd.read_csv("updated_sentiment_data.csv")
concatenated_df_2.head()
```

| | user_name | user_location | user_description | user_created | user_followers | user_friends | us |
|---|---|---|---|---|---|---|---|
| **0** | Irk | Vancouver, WA | Irk started investing in the stock market in 1... | 2018-08-11 03:17:00 | 116.0 | 8.0 | |
| **1** | Xiang Zhang | NaN | Professional Software Engineer ð□□»ð□□□Crypto ... | 2011-01-11 01:37:00 | 42.0 | 22.0 | |
| **2** | Rhizoo | NaN | researcher. local maxima dunningâ□□kruger spec... | 2019-04-03 18:09:00 | 778.0 | 627.0 | |
| **3** | Hari Marquez | Las Vegas, NV | Donâ□□t trust, verify. #Bitcoin \| El Salvador ... | 2014-01-17 23:04:00 | 222.0 | 521.0 | |
| **4** | Bitcoin Candle Bot | Brazil | Robot that posts the closure of the bitcoin da... | 2021-01-06 01:36:00 | 40.0 | 4.0 | |

```
print(concatenated_df_2.columns)
```

```
Index(['user_name', 'user_location', 'user_description', 'user_created',
       'user_followers', 'user_friends', 'user_favourites', 'user_verified',
       'date', 'text', 'hashtags', 'source', 'is_retweet', 'compound', 'score',
       'sentiment_level'],
      dtype='object')
```

```
label_counts = concatenated_df_2['sentiment_level'].value_counts()
print(label_counts)
```

```
Neutral            93170
Positive           35921
Extreme Positive   17344
Negative           15904
Extreme Negative    5316
Name: sentiment_level, dtype: int64
```

```
import sys

# Get the size of the DataFrame in bytes
size_bytes = sys.getsizeof(concatenated_df_2)

# Convert the size to megabytes
size_mb = size_bytes / (1024 * 1024)

# Print the size in MB
print(f"Size of tweets_df: {size_mb} MB")

```

```
Size of tweets_df: 148.0125036239624 MB
```

## Sentiment Analysis with TextBlob

```
#Calculate Polarity and Subjectivity
```

```
import pandas as pd
from textblob import TextBlob


polarity = []
subjectivity = []

# Perform sentiment analysis on each tweet
for tweet in concatenated_df_2['text']:
    try:
        analysis = TextBlob(tweet)
        polarity.append(analysis.sentiment.polarity)
        subjectivity.append(analysis.sentiment.subjectivity)
    except:
        polarity.append(0)
        subjectivity.append(0)

# Add sentiment polarity and subjectivity columns to the dataframe
concatenated_df_2['polarity'] = polarity
concatenated_df_2['subjectivity'] = subjectivity

# Display the updated dataframe
print(concatenated_df_2.head())

```

```
             user_name  user_location  \
0                 Irk  Vancouver, WA
1         Xiang Zhang            NaN
2              Rhizoo            NaN
3        Hari Marquez  Las Vegas, NV
4  Bitcoin Candle Bot         Brazil

                                  user_description         user_created  \
0  Irk started investing in the stock market in 1...  2018-08-11 03:17:00
1  Professional Software Engineer ð»ðCrypto ...  2011-01-11 01:37:00
2  researcher. local maxima dunningâkruger spec...  2019-04-03 18:09:00
3  Donât trust, verify. #Bitcoin | El Salvador ...  2014-01-17 23:04:00
4  Robot that posts the closure of the bitcoin da...  2021-01-06 01:36:00

   user_followers  user_friends  user_favourites  user_verified  \
0           116.0           8.0           4580.0          False
1            42.0          22.0              5.0          False
2           778.0         627.0          32005.0          False
3           222.0         521.0          13052.0          False
4            40.0           4.0              1.0          False

                  date                                               text  \
0  2023-02-25 23:59:00  bitcoin btc rest crypto ye bitcoin cryptocurr ...
1  2023-02-25 23:59:00  retriev invest fund current ongo tidexcoin kic...
2  2023-02-25 23:59:00  bull save monthli thread today good shit bitco...
3  2023-02-25 23:59:00         el salvador shape futur bitcoin membvk32cn
4  2023-02-25 23:59:00  candl day 25022023 close open 2319406 high 232...

                                      hashtags              source  \
0           ['Bitcoin', 'crypto', 'NeedsMoreCrash']     Twitter Web App
1  ['Tidexcoin', 'Kicurrency', 'LMY', 'GMK', 'SYR...  Twitter for iPhone
2                                    ['bitcoin']     Twitter Web App
3                                    ['Bitcoin']     Twitter Web App
```

```
4               ['Bitcoin', 'Candle', 'BearMarket']  Bitcoin Candle Bot

   is_retweet  compound         score sentiment_level  polarity  subjectivity
0         0.0   -0.4019 -2.154092e+05        Negative  0.000000      0.000000
1         0.0    0.0000  0.000000e+00         Neutral  0.000000      0.400000
2         0.0    0.3612  9.005682e+06        Positive  0.250000      0.700000
3         0.0    0.0000  0.000000e+00         Neutral  0.000000      0.000000
4         0.0   -0.2732 -2.240240e+01        Negative  0.053333      0.446667
```

```
concatenated_df_2.to_csv('BitcoinPriceTweets.csv', index=False)

```

## Data For Further Analysis

```
import pandas as pd

# Assuming you have a DataFrame named crypto_usd

# Save the DataFrame as CSV
crypto_usd = pd.read_csv('BitcoinPrice.csv')


```

```
#Crypto - Bitcoin
crypto_usd.head(2)

```

| | time | close | high | low | open | volumefrom | volumeto | Date | Time | |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2023-02-19 13:00:00 | 24682.03 | 24715.82 | 24682.03 | 24707.39 | 903.97 | 22335943.28 | 2023-02-19 | 13:00:00 | 22 |

```
crypto_usd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 337 entries, 0 to 336
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   time         337 non-null    object
 1   close        337 non-null    float64
 2   high         337 non-null    float64
 3   low          337 non-null    float64
 4   open         337 non-null    float64
 5   volumefrom   337 non-null    float64
 6   volumeto     337 non-null    float64
 7   Date         337 non-null    object
 8   Time         337 non-null    object
 9   volume       337 non-null    float64
 10  marketcap    337 non-null    float64
 11  price_delta  336 non-null    float64
dtypes: float64(9), object(3)
memory usage: 31.7+ KB
```

```
#Tweets-Bitcoin
tweets = pd.read_csv('/content/BitcoinPriceTweets.csv')
tweets.head(1)
```

| user_name | user_location | user_description | user_created | user_followers | user_friends | us |
|---|---|---|---|---|---|---|

```
#Dividing the Data in smaller chunks and save it as size of over all data is apporx 140 MB
```

03:17:00

```
import pandas as pd
import os

# Read the dataset from a CSV file
dataset = tweets
# Calculate the desired size of each subset in bytes
desired_size_per_subset = 23 * 1024 * 1024  # Convert 23 MB to bytes

# Calculate the total number of subsets needed
total_subsets = int(round(140 / 23))  # Round up to the nearest integer

# Calculate the number of rows per subset
rows_per_subset = int(round(len(dataset) / total_subsets))  # Round up to the nearest integer

# Create a directory to save the subsets if it doesn't exist
save_directory = '/content'
if not os.path.exists(save_directory):
    os.makedirs(save_directory)

# Split the dataset into subsets and save each subset as a separate CSV file
for subset_index in range(total_subsets):
    start_index = subset_index * rows_per_subset
    end_index = (subset_index + 1) * rows_per_subset
    subset = dataset.iloc[start_index:end_index]

    # Generate the subset file name
    subset_filename = f"BitcoinTweetsPreprocessed_{subset_index + 1}.csv"

    # Generate the full file path
    file_path = os.path.join(save_directory, subset_filename)

    # Save the subset as a CSV file
    subset.to_csv(file_path, index=False)

    # Calculate the size of the subset file
    subset_size = os.path.getsize(file_path)

    # Get the number of rows in the subset
    num_rows = len(subset)

    # Print the name, size, and number of rows of each subset file
    print(f"Subset file: {subset_filename}")
    print(f"Size: {subset_size / (1024 * 1024)} MB")
    print(f"Number of rows: {num_rows}")
    print()

```

```
Subset file: BitcoinTweetsPreprocessed_1.csv
Size: 11.216121673583984 MB
Number of rows: 27942

Subset file: BitcoinTweetsPreprocessed_2.csv
Size: 11.801884651184082 MB
Number of rows: 27942

Subset file: BitcoinTweetsPreprocessed_3.csv
Size: 11.73094654083252 MB
Number of rows: 27942

Subset file: BitcoinTweetsPreprocessed_4.csv
Size: 11.688798904418945 MB
Number of rows: 27942

Subset file: BitcoinTweetsPreprocessed_5.csv
Size: 11.499346733093262 MB
Number of rows: 27942

Subset file: BitcoinTweetsPreprocessed_6.csv
Size: 11.43162727355957 MB
Number of rows: 27942
```

## Datasets for further Anaysis - Preprocessed

```
#Saved the processed file o Github and extracting again for analysis
```

```
#Bitcoin Price
import pandas as pd

# URL to the raw CSV file
url = 'https://raw.githubusercontent.com/Amarpreet3/CIND-820-CAPSTONE/main/Sentimental%20Analysis/BitcoinPricePreprocessed.csv'

# Read the CSV file from the URL
crypto_usd = pd.read_csv(url)

# Display the first few rows of the data
print(crypto_usd.head())


```

```
                  time     close      high       low      open  volumefrom  \
0  2023-02-19 13:00:00  24682.03  24715.82  24682.03  24707.39      903.97
1  2023-02-19 14:00:00  24765.79  24792.85  24679.21  24682.03     1220.29
2  2023-02-19 15:00:00  24928.21  25022.49  24751.96  24765.79     5074.50
3  2023-02-19 16:00:00  24786.44  25175.28  24704.53  24928.21     7094.72
4  2023-02-19 17:00:00  24364.95  24806.64  24346.17  24786.44     6896.84

        volumeto        Date      Time        volume     marketcap  price_delta
0  2.233594e+07  2023-02-19  13:00:00  2.233504e+07  5.512964e+11          NaN
1  3.020300e+07  2023-02-19  14:00:00  3.020178e+07  7.480012e+11        83.76
2  1.263085e+08  2023-02-19  15:00:00  1.263034e+08  3.148644e+12       162.42
3  1.770671e+08  2023-02-19  16:00:00  1.770600e+08  4.388863e+12      -141.77
4  1.693379e+08  2023-02-19  17:00:00  1.693310e+08  4.125910e+12      -421.49
```

```
import pandas as pd

file_urls = [
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_1.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_2.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_3.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_4.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_5.csv',
    'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_6.csv'
]

dfs = []

for url in file_urls:
    # Read the CSV file
    df = pd.read_csv(url)

    # Append the DataFrame to the list
    dfs.append(df)

# Combine all DataFrames into a single DataFrame
combined_df = pd.concat(dfs)

# Display the first few rows of the combined DataFrame
print(combined_df.head())

```

```
             user_name  user_location  \
0                  Irk  Vancouver, WA
1         Xiang Zhang            NaN
2              Rhizoo            NaN
3        Hari Marquez  Las Vegas, NV
4  Bitcoin Candle Bot         Brazil

                                  user_description        user_created  \
0  Irk started investing in the stock market in 1...  2018-08-11 03:17:00
1  Professional Software Engineer ð»ðCrypto ...  2011-01-11 01:37:00
2  researcher. local maxima dunningâkruger spec...  2019-04-03 18:09:00
3  Donât trust, verify. #Bitcoin | El Salvador ...  2014-01-17 23:04:00
4  Robot that posts the closure of the bitcoin da...  2021-01-06 01:36:00

   user_followers  user_friends  user_favourites  user_verified  \
0           116.0           8.0           4580.0          False
```

```
1           42.0          22.0             5.0          False
2          778.0         627.0         32005.0          False
3          222.0         521.0         13052.0          False
4           40.0           4.0             1.0          False

                  date                                               text  \
0  2023-02-25 23:59:00  bitcoin btc rest crypto ye bitcoin cryptocurr ...
1  2023-02-25 23:59:00  retriev invest fund current ongo tidexcoin kic...
2  2023-02-25 23:59:00  bull save monthli thread today good shit bitco...
3  2023-02-25 23:59:00         el salvador shape futur bitcoin membvk32cn
4  2023-02-25 23:59:00  candl day 25022023 close open 2319406 high 232...

                                      hashtags              source  \
0           ['Bitcoin', 'crypto', 'NeedsMoreCrash']     Twitter Web App
1  ['Tidexcoin', 'Kicurrency', 'LMY', 'GMK', 'SYR...  Twitter for iPhone
2                                     ['bitcoin']     Twitter Web App
3                                     ['Bitcoin']     Twitter Web App
4               ['Bitcoin', 'Candle', 'BearMarket']  Bitcoin Candle Bot

   is_retweet  compound         score sentiment_level  polarity  subjectivity
0         0.0   -0.4019 -2.154092e+05        Negative  0.000000      0.000000
1         0.0    0.0000  0.000000e+00         Neutral  0.000000      0.400000
2         0.0    0.3612  9.005682e+06        Positive  0.250000      0.700000
3         0.0    0.0000  0.000000e+00         Neutral  0.000000      0.000000
4         0.0   -0.2732 -2.240240e+01        Negative  0.053333      0.446667
```

```
1 tweets = combined_df.copy()
```

```
1 tweets.head()
```

| | user_name | user_location | user_description | user_created | user_followers | user_friends | us |
|---|---|---|---|---|---|---|---|
| **0** | Irk | Vancouver, WA | Irk started investing in the stock market in 1... | 2018-08-11 03:17:00 | 116.0 | 8.0 | |
| **1** | Xiang Zhang | NaN | Professional Software Engineer ð□□»ð□□□Crypto ... | 2011-01-11 01:37:00 | 42.0 | 22.0 | |
| **2** | Rhizoo | NaN | researcher. local maxima dunningâ□□kruger spec... | 2019-04-03 18:09:00 | 778.0 | 627.0 | |
| **3** | Hari Marquez | Las Vegas, NV | Donâ□□t trust, verify. #Bitcoin \| El Salvador ... | 2014-01-17 23:04:00 | 222.0 | 521.0 | |
| **4** | Bitcoin Candle Bot | Brazil | Robot that posts the closure of the bitcoin da... | 2021-01-06 01:36:00 | 40.0 | 4.0 | |

```
1 print(tweets.columns)
```

```
Index(['user_name', 'user_location', 'user_description', 'user_created',
       'user_followers', 'user_friends', 'user_favourites', 'user_verified',
       'date', 'text', 'hashtags', 'source', 'is_retweet', 'compound', 'score',
       'sentiment_level', 'polarity', 'subjectivity'],
      dtype='object')
```

```
1 import pandas as pd
2
3
4 # Check the shape of the dataset
5 print("Shape of the dataset:", tweets.shape)
6
7 # Check the size of the dataset
```

```
print("Size of the dataset (number of elements):",tweets.size)
```

```
Shape of the dataset: (167652, 18)
Size of the dataset (number of elements): 3017736
```

```
import pandas as pd
import os


# Check the shape of the data
print("Shape of the data:", tweets.shape)
```

```
Shape of the data: (167652, 18)
```

```
label_counts = tweets['sentiment_level'].value_counts()
print(label_counts)
```

```
Neutral             93169
Positive            35921
Extreme Positive    17343
Negative            15903
Extreme Negative     5316
Name: sentiment_level, dtype: int64
```