

▼ Datasets for further Analysis - Preprocessed

1 #Extracting the Preproceed Data for further Analysis

```

1 #Bitcoin Price
2 import pandas as pd
3
4 # URL to the raw CSV file
5 url = 'https://raw.githubusercontent.com/Amarpreet3/CIND-820-CAPSTONE/main/Sentimental%20Analysis/BitcoinPricePreprocessed.csv'
6
7 # Read the CSV file from the URL
8 crypto_usd = pd.read_csv(url)
9
10 # Display the first few rows of the data
11 print(crypto_usd.head())
12
13
```

	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

```

1 import pandas as pd
2
3 file_urls = [
4     'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_1.csv',
5     'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_2.csv',
6     'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_3.csv',
7     'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_4.csv',
8     'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_5.csv',
9     'https://github.com/Amarpreet3/CIND-820-CAPSTONE/raw/main/Sentimental%20Analysis/BitcoinTweetsPreprocessed_6.csv'
10 ]
11
12 dfs = []
13
14 for url in file_urls:
15     # Read the CSV file
16     df = pd.read_csv(url)
17
18     # Append the DataFrame to the list
19     dfs.append(df)
20
21 # Combine all DataFrames into a single DataFrame
22 combined_df = pd.concat(dfs)
23
24 # Display the first few rows of the combined DataFrame
25 print(combined_df.head())
26
```

	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 ...	2018-08-11 03:17:00	
1	Professional Software Engineer & Crypto ...	2011-01-11 01:37:00	
2	researcher. local maxima dunningkruger spec...	2019-04-03 18:09:00	
3	Donâót 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 cndl 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()
```

0	Irk	Vancouver, WA	Irk started investing in the stock market in 1...	2018-08-11 03:17:00	116.0	8.0	4580.0	False	2023-02-25 23:59:00	b re y cry		
1	Xiang Zhang	NaN	Professional Software Engineer ð¤»ð¤»Crypto ...	2011-01-11 01:37:00	42.0	22.0	5.0	False	2023-02-25 23:59:00	retri fun		
2	Rhizoo	NaN	researcher. local maxima dunningâ¬kruger spec...	2019-04-03 18:09:00	778.0	627.0	32005.0	False	2023-02-25 23:59:00	thre !		
3	Hari Marquez	Las Vegas, NV	Donâ¬t trust, verify. #Bitcoin El Salvador ...	2014-01-17 23:04:00	222.0	521.0	13052.0	False	2023-02-25 23:59:00	el sh mer		
4	Bitcoin Candle Bot	Brazil	Robot that posts the closure of the bitcoin da...	2021-01-06 01:36:00	40.0	4.0	1.0	False	2023-02-25 23:59:00	c 2 clk hi		



```
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

```

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

```
9
```

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

```
Size of the dataset (number of elements): 3017736
```

```
1 import pandas as pd
```

```
2 import os
```

```
3
```

```
4
```

```
5 # Check the shape of the data
```

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

```
7
```

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

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

```
2 print(label_counts)
```

```
Neutral 93169
```

```
Positive 35921
```

```
Extreme Positive 17343
```

```
Negative 15903
```

```
Extreme Negative 5316
```

```
Name: sentiment_level, dtype: int64
```

```
1 #Crypto - Bitcoin
```

```
2 crypto_usd.head(2)
```

```
3
```

	time	close	high	low	open	volumefrom	volumeto	Date	Time	volume	marketcap	price_delta
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	22335039.31	5.512964e+11	NaN
1	2023-02-19 14:00:00	24705.70	24702.95	24670.21	24692.02	1220.20	20202001.55	2023-02-19	14:00:00	20201791.26	7.180012e+11	0.27%

```
1 #Tweets-Bitcoin
```

```
2
```

```
3 tweets.head(2)
```

	user_name	user_location	user_description	user_created	user_followers	user_friends	user_favourites	user_verified	date
0	Irk	Vancouver, WA	Irk started investing in the stock market in 1...	2018-08-11 03:17:00	116.0	8.0	4580.0	False	2023-02-25 23:59:00
1	Xiang Zhang	NaN	Professional Software Engineer at Crypto	2011-01-11 01:37:00	42.0	22.0	5.0	False	2023-02-25 23:59:00



▼ Exploratory Data Analysis (EDA)

▼ Data Description

```
1 print(type(df))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
1 df.head(2)
```

	user_name	user_location	user_description	user_created	user_followers	user_friends	user_favourites	user_verified	date
0	Bitcoin Treasures δ□□□	Bitcoin City	I am a bot tracking the USD value of various b...	2020-08-23 16:20:00	2603.0	8.0	48.0	False	2023-03-04 20:00:00 world
1	Bitcoin Price	Order Book	\$BTC price updated every 4 hours\nPrices in ...	2022-03-25 08:31:00	16.0	0.0	2.0	False	2023-03-04 20:00:00 ;

2 rows × 21 columns

1

```

1 import pandas as pd
2
3 # Iterate through each attribute
4 for column in df.columns:
5     unique_count = df[column].nunique()
6     print(f"Unique count for attribute '{column}': {unique_count}")
7

```

```

Unique count for attribute 'user_name': 36053
Unique count for attribute 'user_location': 9432
Unique count for attribute 'user_description': 33094
Unique count for attribute 'user_created': 36531
Unique count for attribute 'user_followers': 11793
Unique count for attribute 'user_friends': 5190
Unique count for attribute 'user_favourites': 17427
Unique count for attribute 'user_verified': 2
Unique count for attribute 'date': 10215
Unique count for attribute 'text': 162425
Unique count for attribute 'hashtags': 40623
Unique count for attribute 'source': 719
Unique count for attribute 'is_retweet': 1
Unique count for attribute 'compound': 1716
Unique count for attribute 'score': 56006
Unique count for attribute 'sentiment_level': 5
Unique count for attribute 'polarity': 2584
Unique count for attribute 'subjectivity': 2594

```

1 df.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 df['sentiment_level'].value_counts()

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

```

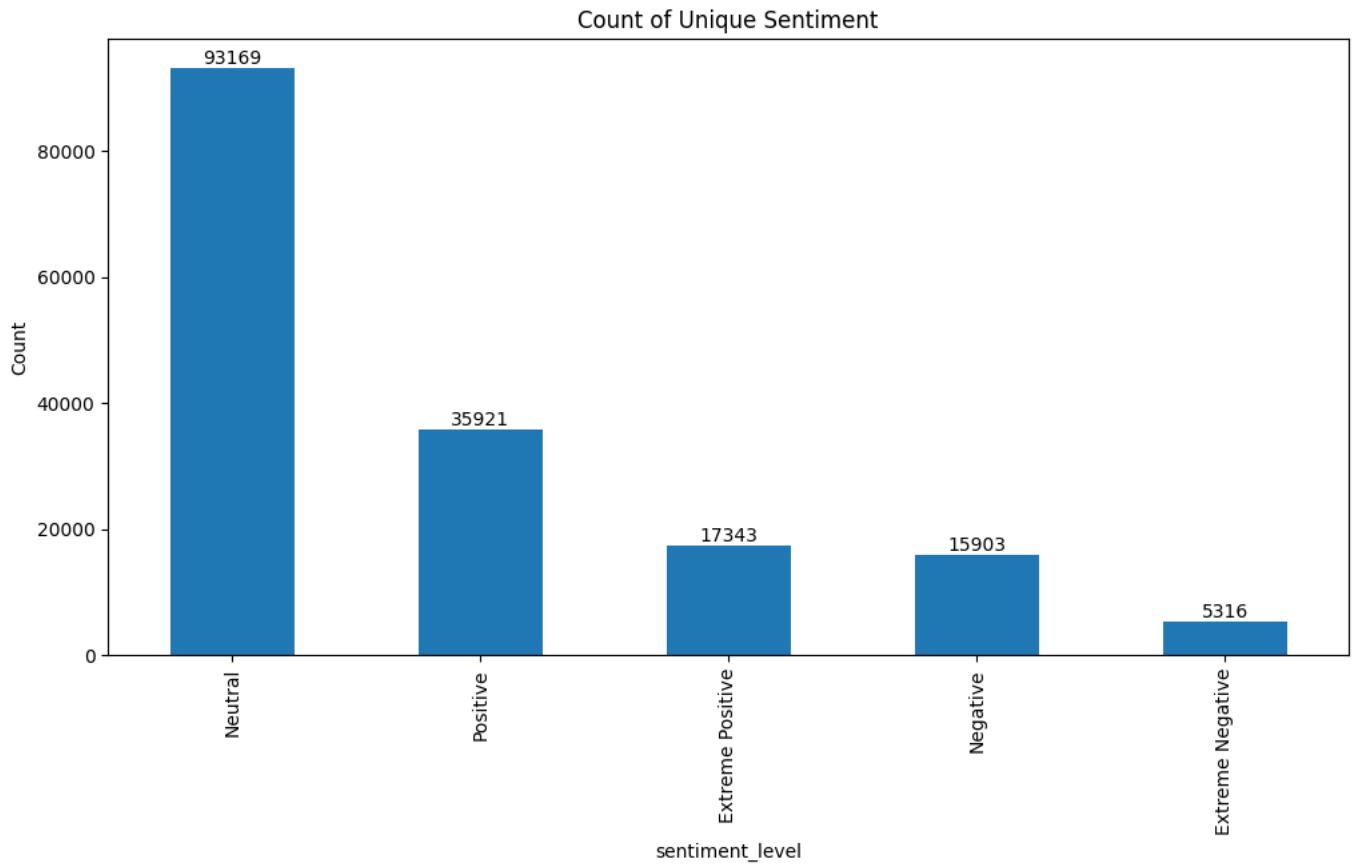
1 import matplotlib.pyplot as plt
2
3 counts = df['sentiment_level'].value_counts()
4
5 # Plotting the bar chart
6 plt.figure(figsize=(12, 6))
7 ax = counts.plot(kind='bar')
8 plt.xlabel('sentiment_level')
9 plt.ylabel('Count')
10 plt.title('Count of Unique Sentiment')
11 plt.xticks(rotation=90)
12

```

```

13 # Labeling the height on top of each bar
14 for p in ax.patches:
15     ax.annotate(str(p.get_height()), (p.get_x() + p.get_width() / 2, p.get_height()), ha='center', va='bottom')
16
17 plt.show()
18

```



```

1
2 print(df.dtypes)

user_name      object
user_location   object
user_description object
user_created    object
user_followers  float64
user_friends    float64
user_favourites float64
user_verified   bool
date           object
text            object
hashtags       object
source          object
is_retweet      float64
compound        float64
score           float64
sentiment_level object
polarity        float64
subjectivity    float64
dtype: object

```

```

1
2 # Get the shape of the merged dataset
3 rows, columns = df.shape
4 print("Number of rows:", rows)
5 print("Number of columns:", columns)

```

Number of rows: 167652
 Number of columns: 18

```

1 # Displaying the summary information
2 summary = df.info()

```

```

3 print(summary)
4

<class 'pandas.core.frame.DataFrame'>
Int64Index: 167652 entries, 0 to 27941
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   user_name        167642 non-null   object  
 1   user_location     83349 non-null   object  
 2   user_description  157323 non-null   object  
 3   user_created      167652 non-null   object  
 4   user_followers    167652 non-null   float64 
 5   user_friends      167652 non-null   float64 
 6   user_favourites   167652 non-null   float64 
 7   user_verified     167652 non-null   bool    
 8   date              167652 non-null   object  
 9   text               167652 non-null   object  
 10  hashtags          167142 non-null   object  
 11  source             167142 non-null   object  
 12  is_retweet        167142 non-null   float64 
 13  compound           167652 non-null   float64 
 14  score              167652 non-null   float64 
 15  sentiment_level   167652 non-null   object  
 16  polarity            167652 non-null   float64 
 17  subjectivity       167652 non-null   float64 
dtypes: bool(1), float64(8), object(9)
memory usage: 23.2+ MB
None

```

```

1 # Let's see meta information about numeric data, we can also see if there any extreme values
2 df.describe()

```

	user_followers	user_friends	user_favourites	is_retweet	compound	score	polarity	subjectivity
count	1.676520e+05	167652.000000	1.676520e+05	167142.0	167652.000000	1.676520e+05	167652.000000	167652.000000
mean	1.069845e+04	774.246069	6.267079e+03	0.0	0.105530	3.260548e+07	0.069989	0.246614
std	1.317029e+05	2691.620939	2.192673e+04	0.0	0.339531	1.440558e+09	0.223690	0.271300
min	0.000000e+00	0.000000	0.000000e+00	0.0	-0.991300	-9.459832e+10	-1.000000	0.000000
25%	1.190000e+02	9.000000	1.100000e+01	0.0	0.000000	0.000000e+00	0.000000	0.000000
50%	5.560000e+02	123.000000	2.810000e+02	0.0	0.000000	0.000000e+00	0.000000	0.223016
75%	1.956000e+03	606.000000	3.300000e+03	0.0	0.361200	7.648344e+03	0.100000	0.418182
max	1.878937e+07	254276.000000	1.083014e+06	0.0	0.984000	1.350093e+11	1.000000	1.000000

```
1 df.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
user_followers	167652.0	1.069845e+04	1.317029e+05	0.000000e+00	119.0	556.000000	1956.000000	1.878937e+07
user_friends	167652.0	7.742461e+02	2.691621e+03	0.000000e+00	9.0	123.000000	606.000000	2.542760e+05
user_favourites	167652.0	6.267079e+03	2.192673e+04	0.000000e+00	11.0	281.000000	3300.000000	1.083014e+06
is_retweet	167142.0	0.000000e+00	0.000000e+00	0.000000e+00	0.0	0.000000	0.000000	0.000000e+00
compound	167652.0	1.055299e-01	3.395308e-01	-9.913000e-01	0.0	0.000000	0.361200	9.840000e-01
score	167652.0	3.260548e+07	1.440558e+09	-9.459832e+10	0.0	0.000000	7648.344025	1.350093e+11
polarity	167652.0	6.998889e-02	2.236899e-01	-1.000000e+00	0.0	0.000000	0.100000	1.000000e+00
subjectivity	167652.0	2.466140e-01	2.712997e-01	0.000000e+00	0.0	0.223016	0.418182	1.000000e+00

▼ Analysing Null/NAN values

```
1 df.isnull().sum()
```

user_name	10
user_location	84303
user_description	10329
user_created	0
user_followers	0

```

user_friends      0
user_favourites   0
user_verified     0
date              0
text              0
hashtags         510
source            510
is_retweet        510
compound          0
score             0
sentiment_level   0
polarity          0
subjectivity      0
dtype: int64

```

```
1 #Null values for attributes which are not significant, so keepin the dataset unaltered
```

```
1 df.isnull().count()
```

```

user_name         27942
user_location     27942
user_description  27942
user_created      27942
user_followers    27942
user_friends      27942
user_favourites   27942
user_verified     27942
date              27942
text              27942
hashtags         27942
source            27942
is_retweet        27942
compound          27942
score             27942
sentiment_level   27942
polarity          27942
subjectivity      27942
dtype: int64

```

▼ Statistics Summary

```

1 # Display summary statistics of numeric columns
2 numeric_columns = df.select_dtypes(include=[int, float])
3 summary_statistics = numeric_columns.describe()
4 print("Summary Statistics:")
5 print(summary_statistics)

```

```

Summary Statistics:
   user_followers  user_friends  user_favourites  is_retweet \
count  1.676520e+05  167652.000000  1.676520e+05  167142.0
mean   1.069845e+04   774.246069   6.267079e+03   0.0
std    1.317029e+05   2691.620939   2.192673e+04   0.0
min    0.000000e+00   0.000000   0.000000e+00   0.0
25%   1.190000e+02    9.000000   1.100000e+01   0.0
50%   5.560000e+02   123.000000   2.810000e+02   0.0
75%   1.956000e+03   606.000000   3.300000e+03   0.0
max   1.878937e+07  254276.000000   1.083014e+06   0.0

   compound       score      polarity  subjectivity
count  167652.000000  1.676520e+05  167652.000000  167652.000000
mean   0.105530  3.260548e+07   0.069989   0.246614
std    0.339531  1.440558e+09   0.223690   0.271300
min   -0.991300 -9.459832e+10  -1.000000   0.000000
25%   0.000000  0.000000e+00   0.000000   0.000000
50%   0.000000  0.000000e+00   0.000000   0.223016
75%   0.361200  7.648344e+03   0.100000   0.418182
max   0.984000  1.350093e+11   1.000000   1.000000

```

▼ Categorical to One-Hot (numeric) Encoding

```

1 #Let's create a list for our categorical columns
2 #cat_cols=["Name", "Symbol", "Date"]

```

```
1 # Create a copy of the data frame in memory with a different name
2 #df_onehot=df.copy()
```

```

3 #convert only categorical variables/features to dummy/one-hot features
4 #df_onehot = pd.get_dummies(df, columns=cat_cols, prefix = cat_cols)
5 #print the dataset
6 #df_onehot

1 #Will do Unigram and Bigram for 'text'

```

▼ Pandas Profiling

```

1 #!pip install pandas-profiling
2

1 #from pandas_profiling import ProfileReport
2 #profile = ProfileReport(df)

1 #profile.to_notebook_iframe()

```

▼ Univariate Analysis

```

1 import matplotlib.pyplot as plt
2 # Univariate analysis for user_created
3 df['user_created'] = pd.to_datetime(df['user_created'])
4 df['user_created_year'] = df['user_created'].dt.year
5 user_created_counts = df['user_created_year'].value_counts().sort_index()
6 user_created_counts.plot(kind='bar')
7 plt.xlabel('User Creation Year')
8 plt.ylabel('Frequency')
9 plt.title('Distribution of User Creation Years')
10 plt.show()
11
12 # Univariate analysis for user_followers
13 user_followers_stats = df['user_followers'].describe()
14 print(user_followers_stats)
15
16 plt.hist(df['user_followers'], bins=20)
17 plt.xlabel('User Followers')
18 plt.ylabel('Frequency')
19 plt.title('Distribution of User Followers')
20 plt.show()
21
22 # Univariate analysis for user_friends
23 user_friends_stats = df['user_friends'].describe()
24 print(user_friends_stats)
25
26 plt.hist(df['user_friends'], bins=20)
27 plt.xlabel('User Friends')
28 plt.ylabel('Frequency')
29 plt.title('Distribution of User Friends')
30 plt.show()
31
32 # Univariate analysis for user_favourites
33 user_favourites_stats = df['user_favourites'].describe()
34 print(user_favourites_stats)
35
36 plt.hist(df['user_favourites'], bins=20)
37 plt.xlabel('User Favorites')
38 plt.ylabel('Frequency')
39 plt.title('Distribution of User Favorites')
40 plt.show()
41
42 # Univariate analysis for user_verified
43 user_verified_counts = df['user_verified'].value_counts()
44 user_verified_counts.plot(kind='bar')
45 plt.xlabel('User Verified')
46 plt.ylabel('Frequency')
47 plt.title('Distribution of User Verification')
48 plt.show()
49
50 # Univariate analysis for date

```

```
51 df['date'] = pd.to_datetime(df['date'])
52 df['date_year'] = df['date'].dt.year
53 date_counts = df['date_year'].value_counts().sort_index()
54 date_counts.plot(kind='bar')
55 plt.xlabel('Year')
56 plt.ylabel('Frequency')
57 plt.title('Distribution of Tweets over Time')
58 plt.show()
59
60 # Univariate analysis for text
61 df['text_length'] = df['text'].apply(lambda x: len(x))
62 text_length_stats = df['text_length'].describe()
63 print(text_length_stats)
64
65 plt.hist(df['text_length'], bins=20)
66 plt.xlabel('Text Length')
67 plt.ylabel('Frequency')
68 plt.title('Distribution of Text Lengths')
69 plt.show()
70
71 # Univariate analysis for hashtags - Will Perform word cloud analysis
```

```
1 # Univariate analysis for source
2 #source_counts = df['source'].value_counts()
```

```
3 #source_counts.plot(kind='bar')
4 #plt.xlabel('Tweet Source')
5 #plt.ylabel('Frequency')
6 #plt.title('Distribution of Tweet Sources')
7 #plt.show()
8000 |
```

1

```
2 # Univariate analysis for is_retweet
3 is_retweet_counts = df['is_retweet'].value_counts()
4 is_retweet_counts.plot(kind='bar')
5 plt.xlabel('Is Retweet')
6 plt.ylabel('Frequency')
7 plt.title('Distribution of Retweets')
8 plt.show()
9
```

10 # Univariate analysis for compound

```
11 compound_stats = df['compound'].describe()
12 print(compound_stats)
13
```

14 plt.hist(df['compound'], bins=20)
15 plt.xlabel('Compound Sentiment Score')
16 plt.ylabel('Frequency')
17 plt.title('Distribution of Compound Sentiment Scores')
18 plt.show()
19

20 # Univariate analysis for score

```
21 score_stats = df['score'].describe()
22 print(score_stats)
23
```

24 plt.hist(df['score'], bins=20)
25 plt.xlabel('Sentiment Score')
26 plt.ylabel('Frequency')
27 plt.title('Distribution of Sentiment Scores')
28 plt.show()
29

30 # Univariate analysis for sentiment_level

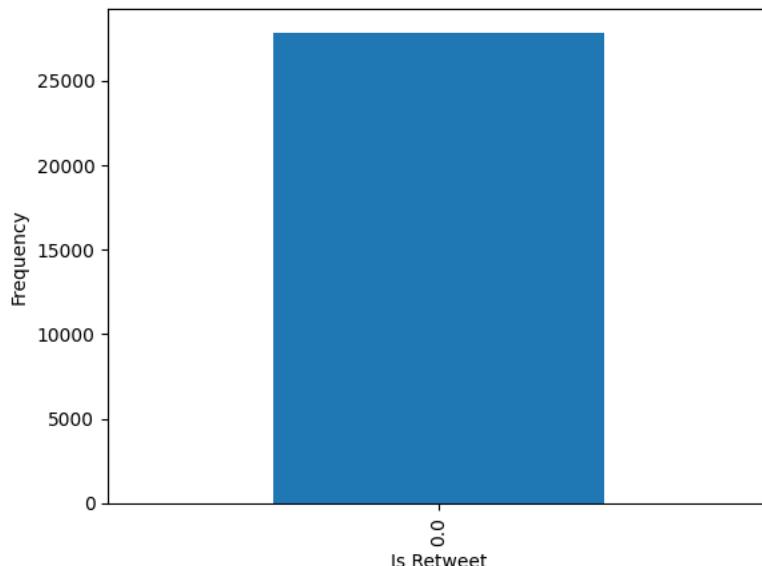
```
31 sentiment_level_counts = df['sentiment_level'].value_counts()
32 sentiment_level_counts.plot(kind='bar')
33 plt.xlabel('Sentiment Level')
34 plt.ylabel('Frequency')
35 plt.title('Distribution of Sentiment Levels')
36 plt.show()
37
```

38 # Univariate analysis for polarity

```
39 polarity_stats = df['polarity'].describe()
40 print(polarity_stats)
41
```

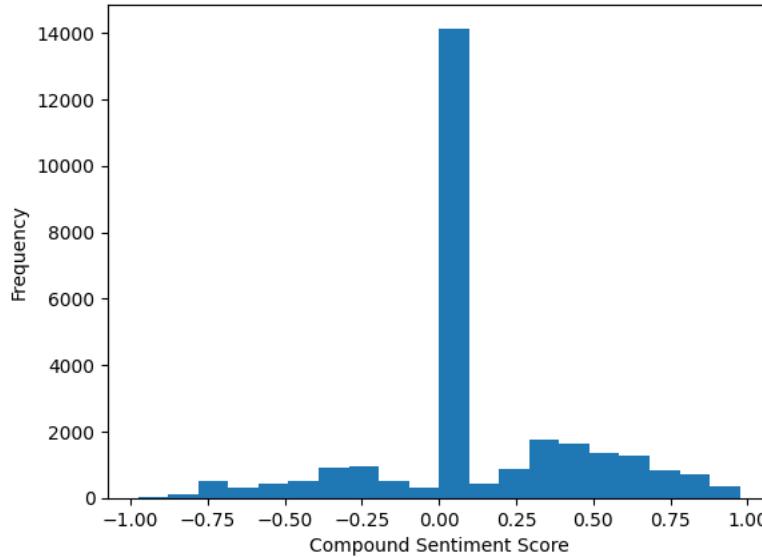
42 plt.hist(df['polarity'], bins=20)
43 plt.xlabel('Polarity')
44 plt.ylabel('Frequency')
45 plt.title('Distribution of Polarity')
46 plt.show()
47

Distribution of Retweets



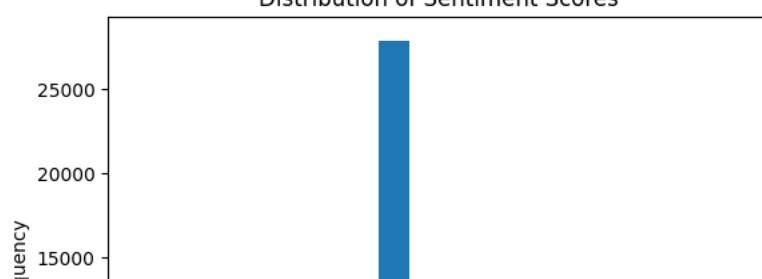
```
count    27942.000000
mean     0.102366
std      0.343487
min     -0.976100
25%     0.000000
50%     0.000000
75%     0.340000
max     0.975300
Name: compound, dtype: float64
```

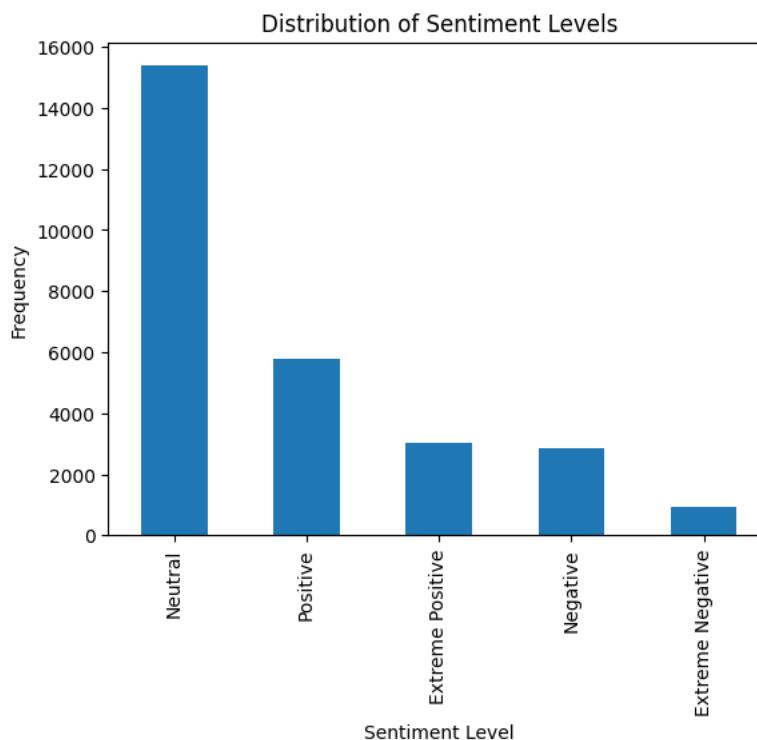
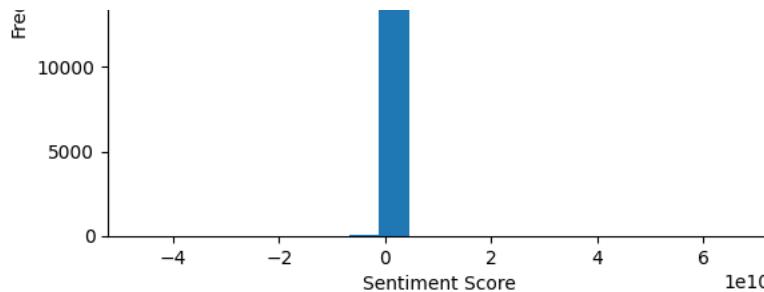
Distribution of Compound Sentiment Scores



```
count    2.794200e+04
mean    2.813255e+07
std     1.346742e+09
min    -4.657105e+10
25%    0.000000e+00
50%    0.000000e+00
75%    8.164702e+03
max    6.683636e+10
Name: score, dtype: float64
```

Distribution of Sentiment Scores





```

count    27942.000000
mean      0.069336
std       0.221637
min     -1.000000
25%      0.000000
50%      0.000000
75%      0.095238
max      1.000000
Name: polarity, dtype: float64

```

Distribution of Polarity



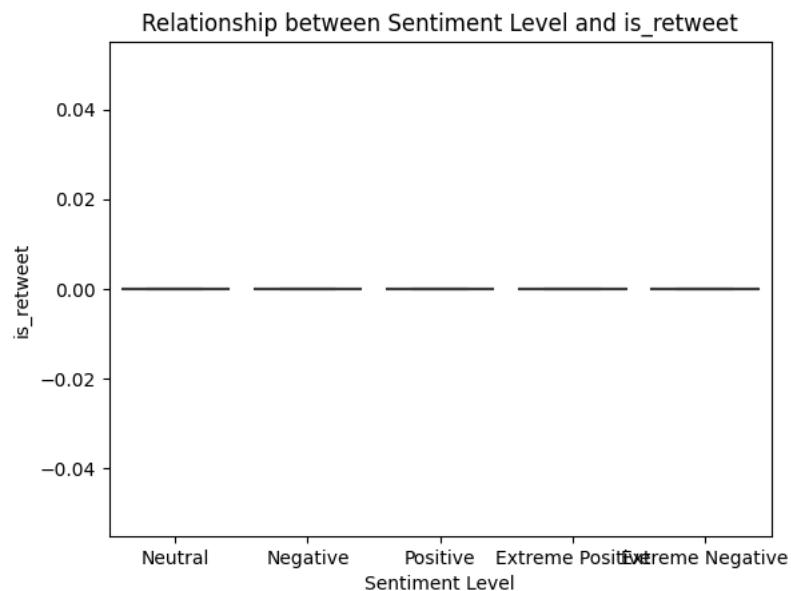
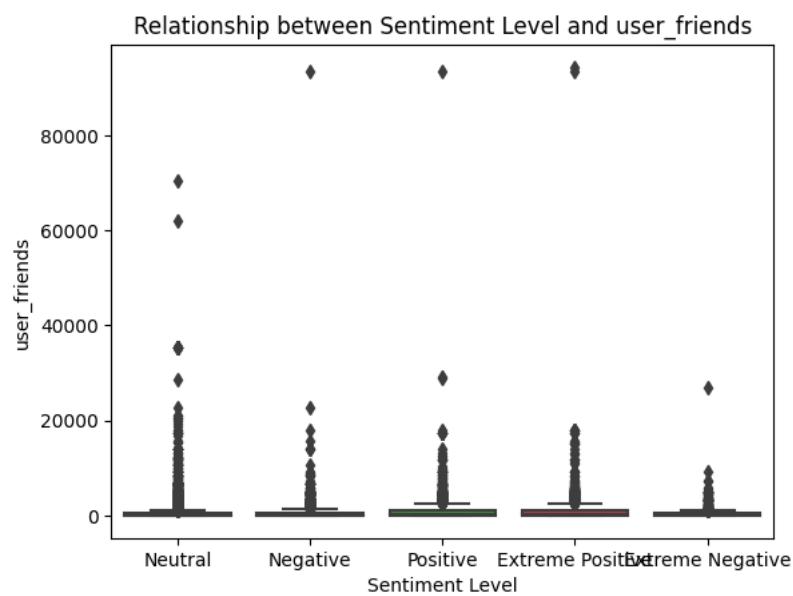
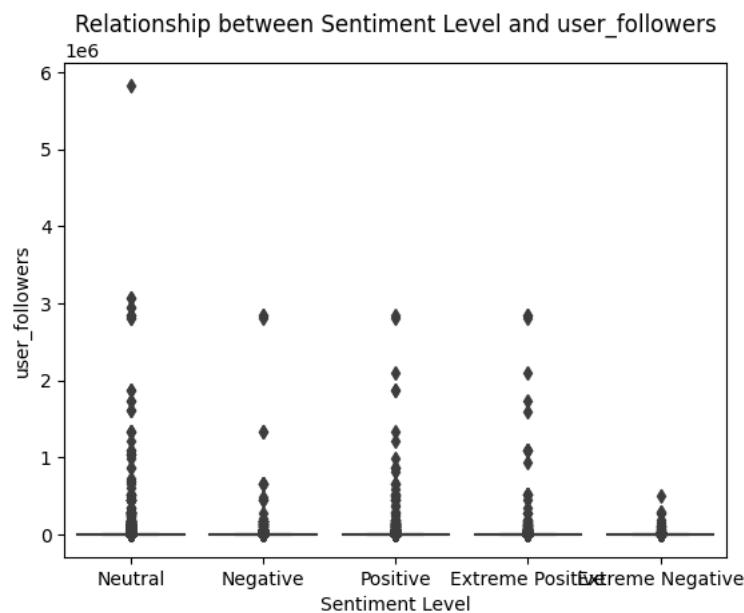
▼ Bivariate Analysis

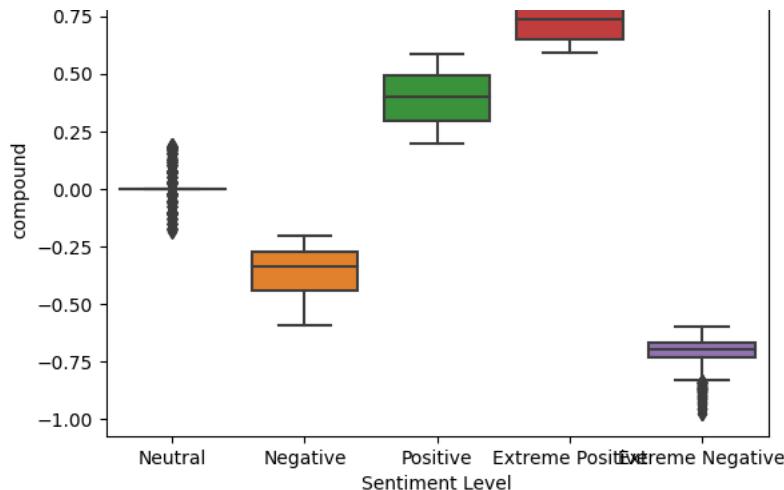
```

1 #Bivariate Analysis of sentiment_level with other attributes
2
3 # Select pairs of attributes for bivariate analysis
4 attribute_pairs = [
5     ('sentiment_level', 'user_followers'),
6     ('sentiment_level', 'user_friends'),
7     ('sentiment_level', 'is_retweet'),
8     ('sentiment_level', 'compound'),
9     ('sentiment_level', 'score'),
10    ('sentiment_level', 'polarity'),
11    ('sentiment_level', 'subjectivity'),
12
13 ]
14
15 # Perform bivariate analysis for each attribute pair
16 for attr_x, attr_y in attribute_pairs:
17     df['sentiment_level'] = df['sentiment_level'].astype(str) # Convert sentiment_level to string type

```

```
18     sns.boxplot(x=attr_x, y=attr_y, data=df)
19     plt.xlabel('Sentiment Level')
20     plt.ylabel(attr_y)
21     plt.title(f'Relationship between Sentiment Level and {attr_y}')
22     plt.show()
23
```





1e10 Relationship between Sentiment Level and score

▼ Data visualization

```
1 !pip install seaborn
2 !pip install matplotlib
3

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.22.4)
Requirement already satisfied: pandas>=0.25 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.5.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in /usr/local/lib/python3.10/dist-packages (from seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (8.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.25->seaborn) (2022.7.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.40.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.22.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (8.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

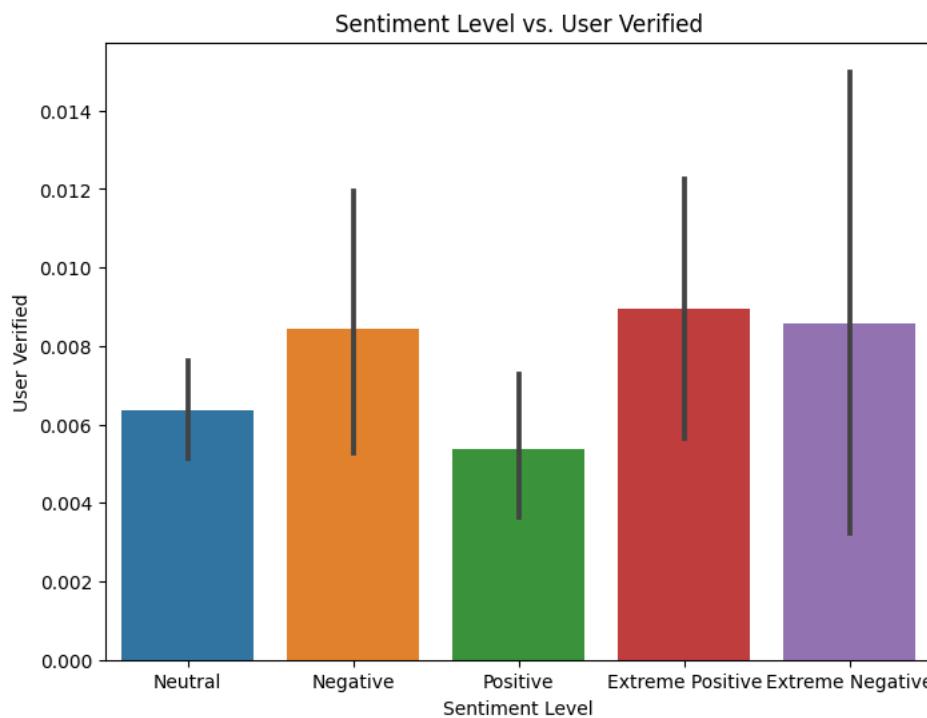
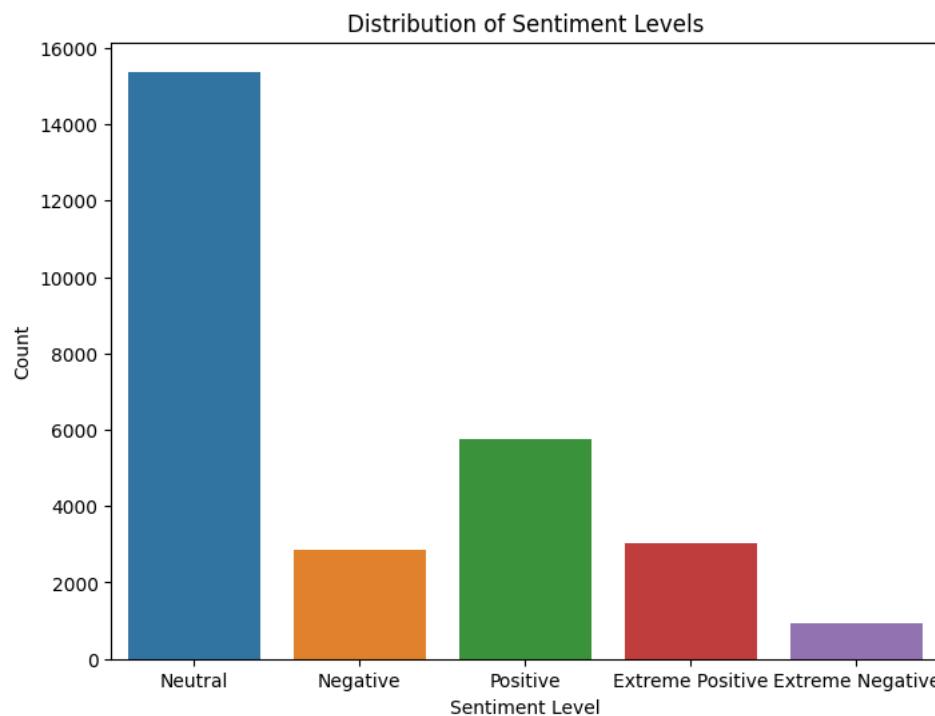
```
1 #Some Visualizations of attributes for analysis
```

```
1 import seaborn as sns
2 import matplotlib.pyplot as plt
3
4 # Scatter plot
5 sns.scatterplot(data=df, x='user_followers', y='user_friends')
6 plt.title('Scatter Plot: User Followers vs User Friends')
7 plt.show()
8
9 # Box plot
10 sns.boxplot(data=df, x='user_verified', y='user_followers')
11 plt.title('Box Plot: User Verified vs User Followers')
12 plt.show()
13
14 # Bar chart
15 sns.countplot(data=df, x='sentiment_level')
16 plt.title('Bar Chart: Sentiment Levels')
```

```
17 plt.show()
18
19 # Histogram
20 sns.histplot(data=df, x='user_favourites', bins=20)
21 plt.title('Histogram: User Favorites')
22 plt.show()
23
24 # Heatmap
25 heatmap_data = df[['user_followers', 'user_friends', 'user_favourites', 'compound', 'polarity', 'subjectivity']].corr()
26 sns.heatmap(heatmap_data, annot=True, cmap='coolwarm')
27 plt.title('Heatmap: Correlation Matrix')
28 plt.show()
29
```

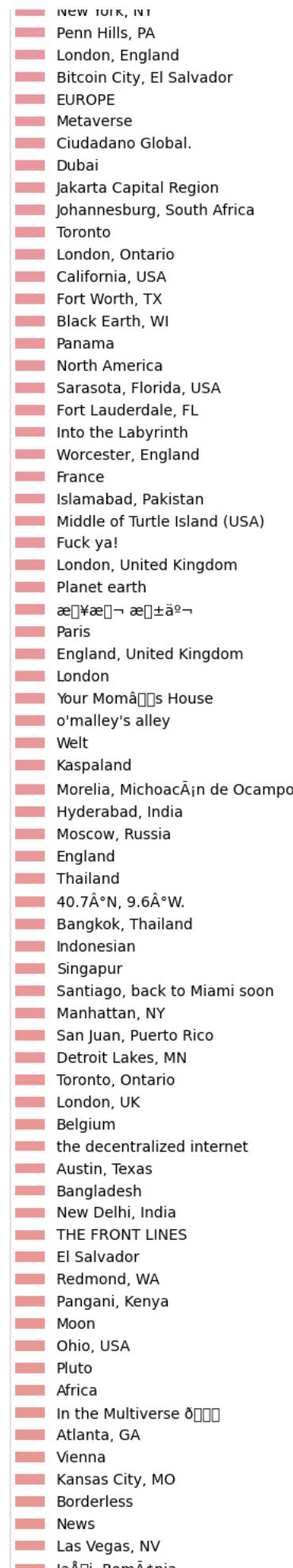
```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4
5 # Convert 'user_verified' column to numeric representation
6 df['user_verified_numeric'] = df['user_verified'].astype(int)
7
8 # EDA for sentiment_level
9
10 # Countplot of sentiment_level
```

```
11 plt.figure(figsize=(8, 6))
12 sns.countplot(x='sentiment_level', data=df)
13 plt.title('Distribution of Sentiment Levels')
14 plt.xlabel('Sentiment Level')
15 plt.ylabel('Count')
16 plt.show()
17
18 # Relationship with other attributes
19
20 # Bar plot of sentiment_level vs. user_verified
21 plt.figure(figsize=(8, 6))
22 sns.barplot(x='sentiment_level', y='user_verified_numeric', data=df)
23 plt.title('Sentiment Level vs. User Verified')
24 plt.xlabel('Sentiment Level')
25 plt.ylabel('User Verified')
26 plt.show()
27
28 # Bar plot of sentiment_level vs. user_location
29 plt.figure(figsize=(12, 6))
30 sns.countplot(x='sentiment_level', hue='user_location', data=df)
31 plt.title('Sentiment Level vs. User Location')
32 plt.xlabel('Sentiment Level')
33 plt.ylabel('Count')
34 plt.legend(title='User Location')
35 plt.show()
36
37 # Scatter plot of sentiment_level vs. user_followers
38 plt.figure(figsize=(10, 6))
39 sns.scatterplot(x='sentiment_level', y='user_followers', data=df)
40 plt.title('Sentiment Level vs. User Followers')
41 plt.xlabel('Sentiment Level')
42 plt.ylabel('User Followers')
43 plt.show()
44
45 # Summary statistics of sentiment_level
46 sentiment_stats = df.groupby('sentiment_level').describe()['user_followers']
47 print("\nSummary Statistics of User Followers by Sentiment Level:")
48 print(sentiment_stats)
49
50 # Remove the temporary column
51 df.drop('user_verified_numeric', axis=1, inplace=True)
52
```



```
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 159 (\x9f) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 145 (\x91) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 137 (\x89) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 151 (\x97) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 156 (\x9c) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 157 (\x9d) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128 (\x80) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 153 (\x99) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 140 (\x8c) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 147 (\x93) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 135 (\x87) missing from current font.
```





02_BitcoinTweets_SentimentAnalysis_EDA.ipynb - Colaboratory



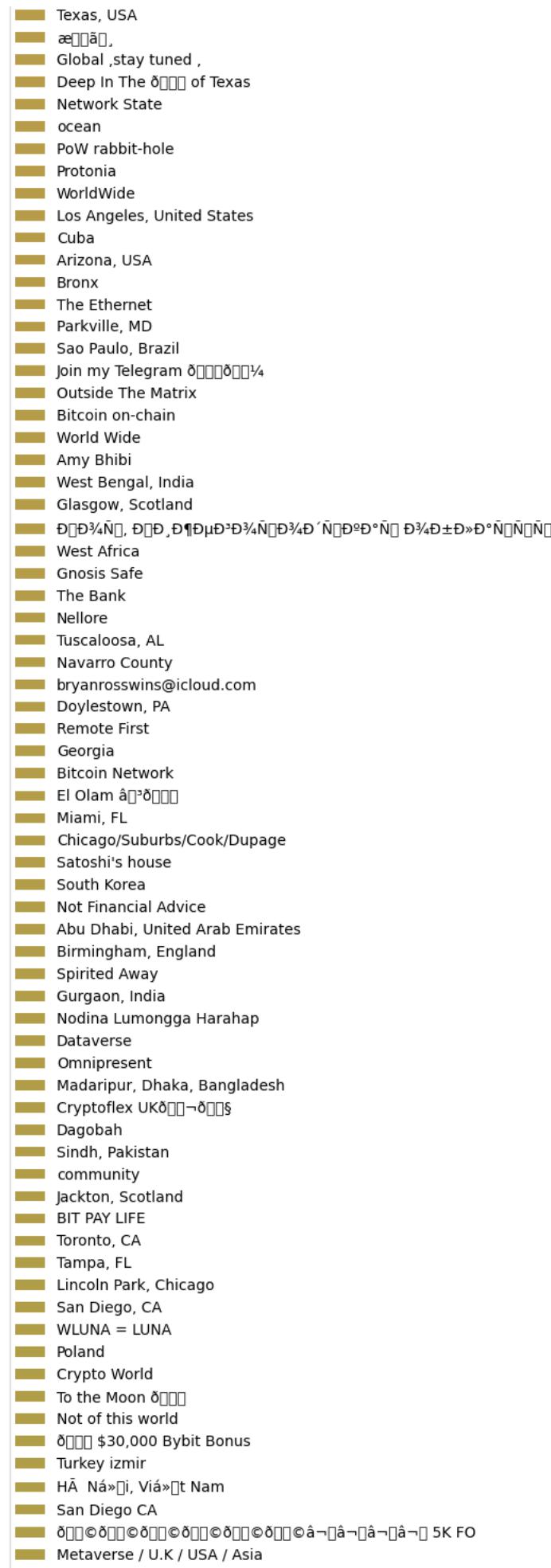




- UK
- Block
- Louisville, KY, USA
- Australia
- New York City
- Washington, DC
- OPSEC
- Amsterdam, Netherlands
- Nevada
- Polygon
- Chicago, IL
- Mind
- Hydra
- Followback Ⓜ️
- The Web
- La Unión, El Salvador
- Queretaro, Mexico
- Ibirapuera - São Paulo- BR
- Sri Mulyani
- Vizille, France
- Blockchain, Bitcoin USA
- Raymore, MO
- Republic of Croatia
- cosmopolitan / planet earth
- hell
- BitcoinⓂ️
- Germany
- Ahmadabad City, India
- Los Angeles, California
- DeFiLand
- everywhere
- Seattle
- West Covina, CA
- Everywhere
- Florida
- GUDIVADA
- Israel
- Lusaka, Zambia
- Ovaa
- Free State of Florida
- bedroom
- Mexico
- Danmark
- Beldanga, India
- The Blockchain Ⓜ️
- Amsterdam, Nederland
- Somewhere in the Middle East
- Barcelona, Spanien
- Amsterdam, The Netherlands
- Mumbai, India
- Dubai, UAE
- www.tradingview.com/u/Daveatt
- MÄ©xico
- MS Physics
- Toronto, Oakville, Canada
- Italy
- Supply & Demand OB
- Đ£Đ°ÑĐ°ÑĐ½Đ°
- Malta Ⓜ️²Đ°
- Southern Illinois
- Karma City
- Sanjauli (Shimla Urban), Shimla
- From Hell to Heaven
- The BlockChain
- worldwide
- Munich, Bavaria
- Buenos Aires, Argentina
- kolkata
- New York, USA
- Bay Area, CA

- Springfield, IL
- Lisboa, Portugal
- Colorado, USA
- Remote
- Check it out
- t.me/investgang
- Southeast Dc
- South Africa
- New Launched Coins
- West Grove, PA
- Evreywhere
- Bitcoin Timechain
- The Blockchain
- ð¤
- Berkeley, Ca
- Imeldaime
- Leipzig, Deutschland
- Planet: Metaverse-GameFi ð¤ ð¤
- Rockies of Colorado.
- Dystopia
- Nigeria, Ukraine (Ð£ÐºÑÐ°ÑÐ½Ð°) +
- Bulgaria
- love people
- Bucaramanga, Colombia
- Ãsterreich
- 2140
- bitcoin.archive.channel@gmail
- Hotlanta
- Email: content@cryptopig.com
- The Pond
- danmark
- Astoria, Queens
- Bitcoin Citadel ð¤-ð¤-
- Malaga
- Mossel Bay, South Africa
- âunker in Fiatnam
- earth
- Down the rabbit hole
- ð¤ð¤
- Warsaw
- Boyd, TX
- Hannover, Deutschland
- 2008
- rabbit hole
- Habiganj, Bangladesh
- On Retest
- Earth
- Citizenship is in the Heavens
- Toledo, OH
- Localhost
- Bengaluru
- 3.1569° N, 101.7123° E
- Ontario
- Charlotte, NC
- Cleveland, GA
- el mundo
- Mountain Cave
- Nashville, TN
- Harmonstown
- metaverse, Europe
- Treaty 6 Territory
- Orlando
- Shrewsbury, England
- Earth ð¤
- Estados Unidos
- cÆVinka
- Rabbit Cave
- A Pale Blue Dot
- Romeð®ð¤¹ / Japanð¤-ð¤

- Greensboro, NC
- Venezuela
- seoul. kangnam
- Washington DC
- ð airdrops for grabs âï¸
- Pretoria, South Africa
- Matrix
- Republic of Serbia
- Catalonia, Spain
- Germany / Canada
- Orlando, FL
- Buitenzorg, Indonesia
- Columbus, OH
- Salt Lake City, UT
- Iran, Mariwann
- Bay Area
- sevilla
- Bitcoin.org
- Girona
- Suffolk, England
- mempool
- Globally
- North Rhine-Westphalia, German
- Untraceable
- The Moon!
- Cyber Space
- Antarctica
- olympus mons
- Turn notifications on!
- Costa Rica
- Lahore, Pakistan
- Quarantine
- IXHUATAN
- Bengaluru (India), Germany
- Decentralised
- The world
- Hopium Resorts
- Bitcoin Blockchain
- Mass NON-COMPLIANCEð
- Los Angeles, Ca
- Jaipur, India
- Detroit, MI
- NFA
- Place with lots of LOVE &FUN
- China
- Fight Club
- Ontario canada
- The Roof
- Earth ð
- NFT, Metaverse ,We love U all
- Barcelona Spain
- Oslo, Norway
- Gunzenhausen
- Indonesia
- Qatar
- Wandering the Cryptospace.
- Metaverse, We love you all !
- From World's Largest Democracy
- JasmyCoin
- Ballito, South Africa
- Black Mesa (BM=BTC Mines)
- IG Farben
- Bull
- Aiming for one \$ Billion AUM
- Near the Beach
- The Right Side of the planet!
- In your thalamus
- I Uber in SLC & Park City, UT
- Houston, TX
- âabylon





- Market
- unknown
- Somewhere in Africa
- Bangalore
- Dubai United Arab Emirates
- Naomi
- Burnaby, British Columbia
- Brooklyn, NY
- Houthalen-Helchteren, Belgia
- Domain Whale 3Digits
- å,-åäöæ°å±åå½æ-å·
- Altavilla Vicentina, Veneto
- Baffin Bay
- Iagos
- Bitcoin 3000000
- korea, è»íè»é¬¼êµ-
- United States of Jalna, India
- Buzz City
- Cryptocurrency
- The Otherside
- Balikesir, Turkey
- Buenos Aires ARGENTINA
- Marne, Germany
- Valais, Switzerland
- Max Prestige
- Otta, Nigeria
- Enchanted Metaworld å“zt3aw.wam
- Seoul, South Korea
- United
- Uzbekistan
- Seoul 30000 / Bangkok 30000-
- Frangokastello
- Lakeland, FL
- your brainwave
- Sintra, Portugal
- Maryland
- Frankfurt, Germany
- Behind you! 30000,0
- Lebanon
- Chicago burbs
- Moorhead, MN
- Orange Coin
- Northern Hemisphere
- Turkey
- Montréal, Quâbec
- Journey to 20k Friends
- Bitcoin Citadel
- Venekistan
- 13th Dimension
- West Virginia, USA
- Illinois, USA
- Finland
- Marion, IA
- CHATTOGRAM , BANGLADESH
- EN MI CASA
- Ampenan
- Ann Arbor, MI
- Atlanta, Georgia
- Pali Rajasthan (India)
- Moldova
- Underground
- Winston - Salem State
- Russia
- 3000000,
- Sumatra
- Sydney, Australia.
- Berlin
- è»íè»é¬¼êµ- i»»ì», Korea
- Ø»»
- Pussay, France

- Colorado
- Casablanca
- Moon.
- JOHANNESBURG (RSA)
- Join 1000+ subscribers ☺
- State College, PA
- Ira Deviyanti Septianingsih
- Overland
- Spaceship to Moon
- South of heaven
- Batangas City, Calabarzon
- Spirit Of Java
- Scarce City
- Nostr Address:
- Tallinn, Estonia
- Hamburg, Deutschland
- Genesis Block
- tor
- New England
- New Hampshire
- Miami, FL
- Jorhat, India
- DLT & ILP, Not Really
- Bucharest, Romania
- Alonzo
- Subscribeð
- Your Mom's house ☺
- Pittsburgh, Pennsylvania
- india
- Mongolia
- Future
- Perth, Australia
- CuraÃ§ao
- Viet Nam
- \$MOON
- Reno, NV
- Bethlehem
- Eindhoven
- michelle qissi
- NEXO PRO Margin Trading
- Midwest USA
- Marcia
- Traveling
- United State
- Join our telegram channel ☺
- Cleveland, OH
- Shenzhen, China
- Innsbruck, Ã¶sterreich
- New Jersey, USA
- nostr:lukedca@noderunners.org
- zBTNshK436
- Wyoming, USA
- No Financial Advices
- Crypto Space
- Active GAsâ€œiï¸ 
- Philadelphia, PA
- Kampala, Uganda
- QuerÃ©taro, QuerÃ©taro Arteaga
- third rock from the sun
- 40 Bank Street. Canary Wharf.
- Bristol, England
- All over the globe
- Somewhere in Asia
- Swindon, England
- AMERICA
- ZÃ¼rich, Schweiz
- Free Market @ Internet Economy
- JOIN FREE TELEGRAM GROUP
- Doctors Office in Metaverse
- humanforever.us





- Always DYOR !!!
- Georgia, Tbilisi
- SPACE
- Inglewood, CA
- Metaverese
- Assam, India
- Tech
- Citadels
- Earn Bitcoin for shopping
- ðµð;
- web3
- Yogyakarta, Indonesia
- Corner of S&P500 and Nasdaq
- Shanghai
- Locust Grove, GA, US, 30248
- Roma, Lazio
- Arin Djunaidi ïé!°
- Argentina
- Join FREE Discord ð
- Haarlem, Netherlands
- EARTH
- Los Altos
- Adana, TÃ¼rkiye
- united States
- NFTð
- Sunny south Florida
- Iceland
- Libertalia
- Sorong
- Global Online
- NFT Community
- I'll follow you back
- cyprus
- @goopdoods
- all
- AntÃ¡rtida
- Take Advantage of my Servicesâ¬
- weltweit
- Middle East
- Block Height 840,000
- The Great Depression
- Manila
- St Louis, MO
- Miami Beach, FL
- UDverse
- Mcdonough, GA
- Christendom
- The Lagoon
- Copacabana, Rio de Janeiro
- St. Louis, MO
- info@cryptopium.com
- Moonzie, Scotland
- Tulsa, OK & Sacramento, CA
- The Islands
- Open & Decentralized Metaverse
- Definately Maybe New Jersey..
- Colorado, US & United Kingdom
- MÃ©rignac, France
- BlockchainCity
- Porto
- Plymouth Colony, Massachusetts
- Poznañ, Polska
- Ordinals
- Community
- In the profit
- Binance Smart Chain
- Bengaluru, India
- United Starts
- Earth (currently)
- â







- Lucerne
- Radix
- metavarse, Nigeria
- Yggdrasil
- Allover
- AQUILAND
- Arnhem, Nederland
- Uganda
- Villain City
- on TPs
- Yola, Nigeria
- MC Capital Ventures
- Portsmouth
- Minato Metaverse
- Colombia
- Butcher Paper
- Kota, Rajasthan
- Dubia UAE
- Check nowâ»
- Palma, Spain
- Samurai Land
- Ponorogo, Indonesia
- Canada / Norway / Australia /
- Telegram @tokensfund
- Budapest, MagyarorszÃ¡g
- Decentralized Planet
- Official YT Channel ððð
- Cypherspace
- ZERO FEES Trading! ð
- Tamilnadu,india
- Galati
- Rajasthan, India
- krishna
- Loxton, South Australia
- Netherland
- Target Moon
- Cryptoworld
- Bavaria Germany
- Italia
- Camden Town, London
- Ferndale
- 5JGL Brno-South, Czechia, 6170
- Calgary, Alberta
- Sydney, Australia
- United states
- BNB Chain, Binance Smart Chain
- Nashville, TN, USA
- Ukraine
- â»â»â»
- Abidjan, Paris
- In the cloudsâ»â»â»
- Peshawar, Pakistan
- Depok
- Cafayate, Argentina
- California, Scotland
- Blockhain
- Suisse
- Guthrie Castle, Scotland, UK
- Madrid, Comunidad de Madrid
- DK
- Middle Earth
- nia minoz
- Ogun, Nigeria
- Italiaðð
- Equatorial Guinea
- Mayfair & France
- Dolarhane
- Nordrhein-Westfalen
- MNW Auto Trader
- Royaume du Maroc

- GURGAON, INDIA
- Otherside CA
- Kota Bandung, Jawa Barat
- ð;ðº
- Dagobah, Outer Rim
- AI TRADING BOT
- Somewhere
- The Moon obvs
- SPOT
- Entering, Valhalla
- Planet ðº
- #blockchain
- On Chain
- Frankfurt am Main
- Beverly Hills, CA
- glorious northyorks
- ðºWeb 3.0
- çåã,
- Caldes de Montbui
- Climbing a mountain.
- Belgrade, Republic of Serbia
- san jose
- Philippines
- Crypto world
- FOR MORE ðº
- Blockchain Everywhere
- Swellendam, South Africa
- Greece - Ukraine
- The Orange Pill Book
- Band Trading IDE
- Wielka Brytania
- TÃ¼rkiye-Amasya
- Lagos, Nigeria
- dexgameevreni
- Newfoundland and Labrador, Can
- Anywhere BTC Accepted
- Blitar - Indonesia
- Defi Index - Insights - Info
- Anonymous
- Blue Mountains, NSW
- blockchain
- Chandigarh, India
- Darfo Boario Terme, Lombardia
- ðº@ðº©ðººðº, ðº~ðººðºº, ðºðººðºº, ðº»ðº©ðººðºº
- Mars - Ø§ÙÙØ±ÙÙØ®
- New York
- Czech republic
- New Orleans, LA
- Subulussalam - Indonesia
- Relai Refcode SWISSTOKEN
- Bangkok (Thailand)
- Sol System
- America
- #Protectcapital ! #Retailarmy
- Michigan
- Curacao E-Zone
- Milano, Lombardia
- Kuusalu vald, Pudisoo kÃ¼la, MÃ¤nnimÃ¤e, 74626, Estonia
- in profits
- Cryptocurrency ðºðºº
- PLR
- Ankara
- Las Vegas
- Plochingen
- Redneck forest
- ðº~ðººðºº!Planet #Earth, mostly
- never mind
- Copenhagen
- #Metaverse
- Anguilla

- not financial advice.
- Luxemburg
- San Juan, Puerto Rico, USA
- Moon, PA
- Luxembourg City
- Ghana
- Country
- La Canada Flintridge, CA
- ð Always DYOR
- Tampa Bay, Florida, USA
- I try not to be a hypocrite
- Anywhere but the Matrix
- Kingston Falls
- Madurai, TamilNadu
- Daddy Island
- Orbis Tertius
- ZÃ¼rich
- Bitcoin Blockchain Block 1
- Pune (India)
- Calabar, CR, Nigeria
- Ù¾Ø§Ù©Ø³Ø³Ø§Ù
- Gods own city
- Marbella
- Pennsylvania, USA
- Utrecht
- Lucknow, India
- instagram:1571casin0
- Berlin, Germany
- Berlin, London, New York
- Dominica
- Anggun Lestari UAK
- Bitcoin Heaven
- Uyo Nigerian
- 1990s INTERNET is CRYPTO NOW
- Slovenia
- BSC
- Tasmania, Australia
- Everywhere and No Where
- Bitcoin World
- Heidelberg, Germany
- Somewhere in Matrix
- Gujarat India.
- International Space Station ð
- HongKong
- limbo
- Fife, Scotland.
- Silicon Valley, Carlifornia,
- âlieuâ
- Crypto
- Tanzania
- Colorado Springs, CO
- Minneapolis, MN
- Mempool
- Winterthur, Switzerland
- Kentucky
- ðð
- 32000
- Indore, India
- Land of the Free
- Buy low, sell high
- A small blue dot in space
- Twetch @63757
- silviani
- kuwait
- ðð±
- GigaMegaTurboStonks
- The Metaverse
- New-York
- Goa, India
- london, kensington



simulator 0_0
 Birbhum WEST BENGAL INDIA
 Between Metaverse & Earth
 δ_0_0
 m/84'/0'/0'/0/0
 Claimδ_0_0 Voucher 100USDT
 Āoge
 Mint Island
 Kolkata, India
 7 moves ahead.
 Worlwide
 Coronado, CA
 Virginia, USA
 Redwood City, California
 Sky
 Metaverse δ_0_0
 Ontario, Canada δ_0_0 δ_0_0
 Digital
 Running Springs
 The Antares Star System
 Pleadies
 Orland Park, IL
 UTXO
 Palermo, Sicily
 Hafizabad, Pakistan
 Capric-Anna.nft
 Turn the δ_0_0 NOTIFICATIONS ON
 in the pines
 Crypto Twitter
 Somewhere in the MetaVerse
 Austin TX
 The Promised Land
 Grovetown, GA
 Crypto | Forex | Stocks
 Toronto, Canada
 internet
 CRYPTO WOLD
 Montrâal, Canada
 δ_0_0;δ_0_0δ_0_0PfknRδ_0_0δ_0_0δ_0_0δ_0_0
 Somewhere on Earth
 Maryland, US
 Kolkata
 Maui, Hawaii
 California
 DKI Jakarta, Indonesia
 Jericoacoara - CE
 Spirit Realm
 Spokane - Coeur D Alene
 Bonke world
 EL SALVADOR
 Nashville
 Vientiane, Laos / Canada
 Republic of the Philippines
 Minnesota
 21M Block 0101010101010101010101
 Orange County
 Gold Coast ,Australia
 Rossland, British Columbia
 Kenya, Uganda
 Temecula, CA
 Security beyond cold storage.
 TWITTER ALPHA
 AnyGalaxy
 Detroit
 Delhi
 Bethlehem, PA
 Denpasar, Bali
 Canada â_0_0 China
 Britain
 Laren, Netherlands
 Brno, Czech Republic

- Prague, Czech Republic
- Cork, Ireland
- The Metaverse and Web3
- Bavaria, Germany
- pakiatan
- Always Traveling
- Off-world
- da blockchains & wassieverse
- Mafia Secret Room
- â¬tcoin Ãtitadel
- Singapore, US and Hong Kong
- unusual whales
- On my bed
- Alger
- 3100 E Charleston Las Vegas NV
- National Capital Region, Repub
- Join our Discord ð
- Location
- Sylvester, GA
- Klaten Selatan, Indonesia
- All Over The World
- ðð, ð'ð'ð'ð·ðµðâ
- Maki's Izakaya - Land of Wano
- New York City / Jersey City
- Virginia
- Inscribed on Bitcoin
- The Lone Star State
- Sydney, New South Wales
- Ã°nternational
- Cash is Trash
- Analog, Digital, Space
- Hey! ð
- \$BITCOIN
- ð Review Bitcoin Products
- Get Binance
- Naples, Williamstown, Anghiari
- Magelang, Jawa Tengah
- Orion-Cygnus Arm
- Alternate Universe #KX6M8
- Chiyoda-ku, Tokyo
- Ceilter of the world
- Wall Street
- Melbourne, FL
- LatinoamÃ©rica
- Winnipeg
- Vermont
- Wyoming, USA, Thayne
- Bitcoin MagazineÂ®
- ðð;
- My mom's house
- Sunyani
- This account is not monitored
- https://www.btcteacher.com
- 21 Bitcoin Avenue, Mars
- Gold Coast, Australia
- Tel Aviv, Israel
- Kansas
- AlgodÃ±o
- Santiago, Chile
- San Antonio CaÃ±ada, Puebla
- Fairfield Township Ohio
- Lowell, MA
- Casino Royale
- Alberta, Canada
- Worldðwide
- Land of Satoshi
- St. John's
- San Diego, CA. 92109
- Sioux Falls, SD
- Kullu and Manali, India

2045
 North Dakota
 Cascadia
 Bitcoiner
 NO TELEGRAM / NO DISCORD
 I appreciate your support ðð»
 nowhere
 Heading to El Salvador
 â
 Wildwood, MO
 Freemount
 #soon VeVeVerse
 SÃ£o Paulo
 Arkansas
 CNFT
 bitcoin
 Brisbane, AU ðð
 Bitcoin & Markets
 ANKARA
 Vancouver, British Columbia
 Wasteland
 Central Java, Indonesia
 Juno Network
 U.S
 Otherside
 BTCSUD
 officialcryptoposters.com
 India(World Largest Democracy)
 Whale Brain
 Nellore, Andhra Pradesh
 Australia ,N.S.W
 Cryptometaverse
 Coweta, Ok
 Level Up & Learn
 Baltimore, MD
 33 Liberty St, New York, NY
 Samford, CT
 Galveston Island, TX
 Washington, USA
 MARS ðð
 https://gm.co/seller/46
 Brisbane, Queensland
 Celestis - City of the Gods
 Santa Tecla/Bitcoin Beach
 Edmonton, Alberta
 Little Rock, AR
 Santiago Metropolitan Region,
 Layer 1 & 2.
 Philly
 Cambridge, MA
 Tyler, TX
 Bangkok
 US
 Victoria, British Columbia
 ðð»
 moscow
 ê½ê°ë¶ëµ±ë¶ëµë¶ëµ ë¶ëµêµ¬
 Province of Moscow, Russia
 Satoshis basement
 Yokohama City, Kanagawa
 In The Archives âí,âí
 Auckland, New Zealand
 Bitcoin Education Discord
 Somewhere, Someplace
 Missouri
 Deep in the Metaverse
 Chicago,IL
 Pensacola
 Jericoacoara / Bitcoin Beach
 Universe GN-z11





