

# Contents

Introduction	3
Dataset and Source	3
Part 1: Analysis of tweets with hashtag	3
Part 2: Extract tweets by Elon Musk for analysis	6
Part 3: Sentiment Analysis	8
Program Description	13

### Introduction

The project is an attempt to extract data and analyze data from Twitter using python library Tweepy. Tweepy is an open-source Python package that gives you a very convenient way to access the Twitter API with Python. Tweepy includes a set of classes and methods that represent Twitter's models and API endpoints, and it transparently handles various implementation details.

The IST Mini project-2 is an attempt to extract and analyze semi-structure data from Twitter using Tweepy. The project is divided in three minor parts namely:

1. Extracting and analyzing tweet's containing twitter Hashtag

- 2. Analyzing a users' tweets and finding meaningful statistics
- 3. Sentiment analysis using TextBlob

### Dataset and Source

The Dataset is in JSON format. I have utilized tweepy Cursor interface and user timeline method to extract and iterate through tweets in python.

## Part 1: Analysis of tweets with hashtag

In this part, I used '#covid19' search query to fetch 200 tweets from twitter. Thereafter, I initialized pandas' data frame to populate each tweet in it. I used 8 attributes from the twitter object namely - user\_name, user\_location, user\_description, user\_verified, date, text, hashtags, and source.

```
# populate the dataframe
    for tweet in tweets copy:
       hashtags = []
       try:
            for hashtag in tweet.entities["hashtags"]:
                hashtags.append(hashtag["text"])
           text = api.get_status(id=tweet.id, tweet_mode='extended').full_text
       except:
            pass
       tweets_df = tweets_df.append(pd.DataFrame({'user_name': tweet.user.name,
                                                   'user_location': tweet.user.location,\
                                                   'user_description': tweet.user.description,
                                                   'user_verified': tweet.user.verified,
                                                   'date': tweet.created at,
                                                   'text': text,
                                                   'hashtags': [hashtags if hashtags else None],
                                                   'source': tweet.source}))
       tweets_df = tweets_df.reset_index(drop=True)
```

#### tweets df.head()

- Above command is used to view the top 5 tweets in the data frame

The dimension of data frame is calculated using the shape command as shown below: -

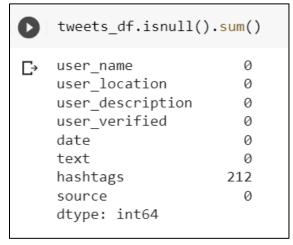
```
tweets_df.shape
(200, 8)
```

tweets\_df['user\_verified']. value\_counts()

- Above command is used to get the pivot table for summary of verified accounts.

```
[342] tweets_df['user_verified']. value_counts()

False 436
True 64
Name: user_verified, dtype: int64
```



Check for null values using isnull method.

```
tweets_df.user_location.value_counts()
Ľ→
                                     120
    Los Angeles, CA
                                      75
    Manila, Philippines
    New Delhi, India
    Manhattan, NY
    Hamilton City, New Zealand
                                       1
    Toronto, Canada
    Virginia, USA
                                       1
    Toronto
                                       1
    Unlit Lowlands, Plague Island
    Name: user location, Length: 212, dtype: int64
```

Check for value counts on location attribute suggests that most number of tweets in the data frame are from California, followed by Philippines.

### Part 2: Extract tweets by Elon Musk for analysis

In this part, I extracted 200 tweets by Elon Musk using user\_timeline () method of tweepy, and then iterated through each tweet to add them to list.

```
[325] import pandas as pd
    df = pd.DataFrame(list)
    df.to_csv('refined_tweets.csv')
    df.shape
    (200, 4)
```

As seen above, the data frame dimension is 200 rows and 4 columns. Columns are namely text, favorite\_count, retweet\_count, and created\_at.

Created\_at column is a time stamp. We converted it to datetime type of pandas and extracted date from the same.

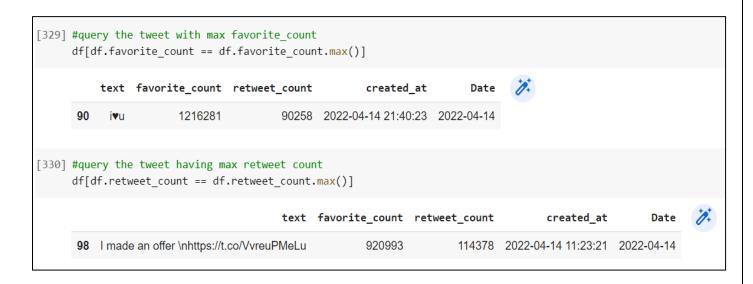
```
#convert to pandas date-time
df['created_at'] = pd.to_datetime(df['created_at'])

# remove time from Date and store it in a new column
df['Date'] = df['created_at'].dt.date
# display the dataframe
print(df)
```

Number of tweets per month are calculated with values counts command as shown below

```
#number of tweets by date
df.Date.value_counts()
2022-04-21
2022-04-09
2022-04-03
             19
2022-04-08
             16
2022-04-18
             15
             14
2022-04-14
2022-04-10
             13
2022-04-16
2022-04-20
2022-04-07
2022-04-05
             10
2022-04-06
             7
2022-04-04
2022-04-15
2022-04-17
2022-04-19
2022-04-22
Name: Date, dtype: int64
```

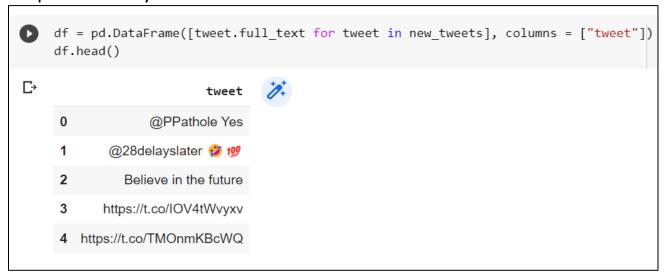
The final business questions are to query the tweet with the most favorite count and the tweet with most retweet count.



As shown above, we use max () function to answer these business queries.

## Part 3: Sentiment Analysis

At first, only the tweet message is loaded into the data frame to make things simpler for analysis.

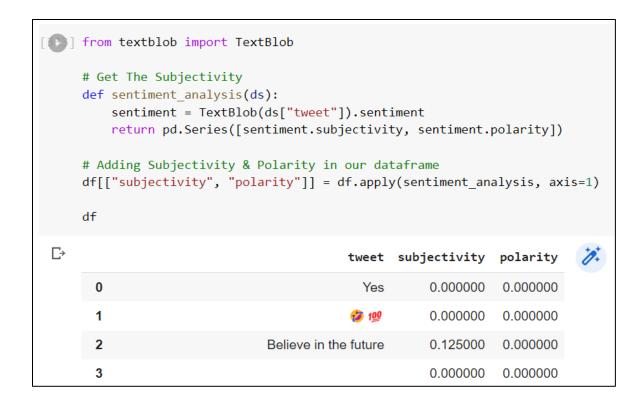


The next process is to clean the data. i.e., to remove hashtags, mentions, retweets, links etc.

```
# Clean The Data
def cleantext(text):
    text = re.sub(r"@[A-Za-z0-9]+", "", text) # Remove Mentions
    text = re.sub(r"#", "", text) # Remove Hashtags Symbol
    text = re.sub(r"RT[\s]+", "", text) # Remove Retweets
    text = re.sub(r"https?:\/\/\S+", "", text) # Remove The Hyper Link
    return text

# Clean The Text
df["tweet"] = df["tweet"].apply(cleantext)
```

After this, we calculate subjectivity and polarity to understand the sentiment of a tweet.



Further, we create a word cloud to find the most frequent and relevant word used in the text message.

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud

allwords = " ".join([twts for twts in df["tweet"]])
wordCloud = WordCloud(width = 1000, height = 1000, random_state = 21, max_font_size = 119).generate(allwords)
plt.figure(figsize=(20, 20), dpi=80)
plt.imshow(wordCloud, interpolation = "bilinear")
plt.axis("off")
plt.show()
```

We use categorization to point as tweet as neutral, negative, or positive depending on the calculated score as shown below.

```
def analysis(score):
   if score < 0:
        return "Negative"
    elif score == 0:
       return "Neutral"
    else:
       return "Positive"
# Create a New Analysis Column
df["analysis"] = df["polarity"].apply(analysis)
# Print The Data
df
                                      tweet subjectivity polarity analysis
 0
                                                  0.000000
                                                            0.000000
                                         Yes
                                                                         Neutral
                                                  0.000000
                                                             0.000000
                                                                         Neutral
 1
                                       🤣 1<u>00</u>
                           Believe in the future
 2
                                                  0.125000
                                                             0.000000
                                                                         Neutral
```

Finally, positive, and negative tweets are highlighted and a ratio of length of positive tweets to negative tweets is calculated to comment on the sentiment analysis of a user's tweet.

```
positive_tweets = df[df['analysis'] == 'Positive']
negative_tweets = df[df['analysis'] == 'Negative']

print('positive tweets')
for i, row in positive_tweets[:5].iterrows():
    print(' -' + row['tweet'])

print('negative tweets')
for i, row in negative_tweets[:5].iterrows():
    print(' -' + row['tweet'])
```

```
[337] len(positive_tweets) / len(negative_tweets)
4.238095238095238

We can conclude that as the ratio is +ve, the overall tweets by user Elon Musk are positive in nature
```

## Program Description

Below attached is the Jupyter notebook consisting of all the executable commands in Python. I have added the text blocks in each command to better understand the dynamics of each code block.

