Effect of Twitter on the stock market

- This dataset contains over 3 million unique tweets with their information such as tweet id, author of the tweet, post date, the text body of the tweet, and the number of comments, likes, and retweets of tweets matched with the related company.
- we will find if the tweets related to a company affect its stock price

Data cleaning and organising

importing libraries

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats as stat
from pylab import rcParams
```

taking a look at the data available

In [ ]:
```
comp=pd.read_csv('datasets/effect of twitter/Company.csv')
comp_tweet=pd.read_csv('datasets/effect of twitter/Company_Tweet.csv')
tweet=pd.read_csv('datasets/effect of twitter/Tweet.csv')
```

In [ ]:
```
comp
```

In [ ]:
```
comp_tweet.head(5)
```

In [ ]:
```
tweet.head(5)
```

merging datasets

merging tweet and comp_tweet to identify the company in the tweet dataset

In [ ]:
```
tweets=pd.merge(tweet,comp_tweet,on='tweet_id',how='inner')
tweets.head(5)
```

formatting post_date column

In [ ]:
```
tweets['post_date'] = pd.to_datetime(tweets['post_date'], unit='s')
tweets['date'] = pd.to_datetime(tweets['post_date'].apply(lambda date: date
```

In [ ]:
```
tweets['date'].head()
```

removing post date colum

In [ ]:
```
tweets.drop('post_date',axis=1,inplace=True)
```

information about our dataset

In [ ]:
```
tweets.info()
```

dealing with null values

checking for null values

In [ ]:
```
tweets.isna().sum()
```

replacing null values in writer column to anonymous

In [ ]:
```
tweets['writer'] = tweets['writer'].fillna('anonymous')
```

how many tweets were made about each company

In [ ]:
```
tweets.ticker_symbol.value_counts()
```

- GOOGL is with voting rights and GOOG is without.
- We only consider stocks with voting rights for our analysis

time-period of tweets

In [ ]:
```
start_date=min(tweets['date'])
end_date=max(tweets['date'])
print(start_date,'\n',end_date)
```

- the first tweet was recorded on 1-1-2015 and the last was on 31-12-2019
- we have 5 years worth of data
- these are our start and end dates

creating seperate dataframes for each company

```
In [ ]: aapl = tweets[tweets['ticker_symbol'] == 'AAPL']
        tsla = tweets[tweets['ticker_symbol'] == 'TSLA']
        amzn = tweets[tweets['ticker_symbol'] == 'AMZN']
        msft = tweets[tweets['ticker_symbol'] == 'MSFT']
        googl = tweets[tweets['ticker_symbol'] == 'GOOGL']
```

getting stock prices dataset

```
In [ ]: !pip install yfinance -q
```

```
In [ ]: import yfinance as yf  # yahoo finance
```

TESLA

```
In [ ]: tsla_stock = yf.Ticker('TSLA')
```

```
In [ ]: tsla_stock = tsla_stock.history(start= min(tsla['date']), end=max(tsla['dat
```

```
In [ ]: tsla_stock.index=tsla_stock.index.date
        tsla_stock['date']=tsla_stock.index
        tsla_stock['date']=tsla_stock['date'].apply(pd.to_datetime)
        tsla_stock.head(2)
```

APPLE

```
In [ ]: aapl_stock = yf.Ticker('AAPL')
        aapl_stock = aapl_stock.history(start= min(aapl['date']), end=max(aapl['dat
        aapl_stock.index=aapl_stock.index.date
        aapl_stock['date']=aapl_stock.index
        aapl_stock['date']=aapl_stock['date'].apply(pd.to_datetime)
        aapl_stock.head(2)
```

AMAZON

```
In [ ]: amzn_stock = yf.Ticker('AMZN')
        amzn_stock = amzn_stock.history(start=min(amzn['date']), end=max(amzn['date
        amzn_stock.index = amzn_stock.index.date
        amzn_stock['date'] = amzn_stock.index
        amzn_stock['date'] = amzn_stock['date'].apply(pd.to_datetime)
        amzn_stock.head(2)
```

GOOGLE

In [ ]:
```
googl_stock = yf.Ticker('GOOGL')
googl_stock = googl_stock.history(start=min(googl['date']), end=max(googl['
googl_stock.index = googl_stock.index.date
googl_stock['date'] = googl_stock.index
googl_stock['date'] = googl_stock['date'].apply(pd.to_datetime)
googl_stock.head(2)
```

MICROSOFT

In [ ]:
```
msft_stock = yf.Ticker('MSFT')
msft_stock = msft_stock.history(start=min(msft['date']), end=max(msft['date
msft_stock.index = msft_stock.index.date
msft_stock['date'] = msft_stock.index
msft_stock['date'] = msft_stock['date'].apply(pd.to_datetime)
msft_stock.head(2)
```

ANALYSIS

To find out if amount of tweets affects the volume traded of the company

defining fuction for plotting

In [ ]:
```
def tweet_vol(tweet,stock,title):
    md2=pd.merge(tweet, stock, on='date', how='inner')
    tweet_volume =md2.groupby('date').size().rolling(30).mean().dropna()
    stock_volume =stock['Volume'].rolling(30).mean().dropna()
    corr1=tweet_volume.corr(stock_volume)
    print("coorelation is: ",corr1)
    fig, ax = plt.subplots()
    ax.plot(tweet_volume,color='orange',label='tweets')
    ax2 = ax.twinx()
    ax2.plot(stock_volume,label='stock')
    plt.title(title)
    ax.set_xlabel('year')
    ax.set_ylabel('tweet vol')
    ax2.set_ylabel('stock vol')
    ax.legend()
    ax2.legend(loc='upper left')
    plt.show()
```

In [ ]:
```
plt.rcParams['figure.figsize'] = (15, 5)
tweet_vol(tsla,tsla_stock,'tesla')
```

In [ ]:
```
tweet_vol(aapl,aapl_stock,'apple')
```

In [ ]:
```
tweet_vol(amzn,amzn_stock,'amazon')
```

In [ ]:
```
tweet_vol(googl,googl_stock,'google')
```

In [ ]:
```
tweet_vol(msft,msft_stock,'microsoft')
```

interpretation

- the sheer volume of tweets has a correlation with the trade volume.
- all compaines except for microsoft have a moderate positive correlation between volume of tweets and stocks

Sentinent Analysis

classifying positive and negative tweets and their affect on stock prices

we choose the top 1,00,000 tweets based on the number of likes and evalute the sentiment on the basis of those.

In [ ]:
```
def most_liked_tweets(twee):
    twee = twee.sort_values(by=['like_num'], ascending=False)
    twee=twee.iloc[:100000]
    return twee
```

In [ ]:
```
tsla_ml=most_liked_tweets(tsla)
aapl_ml=most_liked_tweets(aapl)
amzn_ml=most_liked_tweets(amzn)
googl_ml=most_liked_tweets(googl)
msft_ml=most_liked_tweets(msft)
```

text processing

removing hyperlinks, special charecters and numbers and converting to lower case

In [ ]:
```
import nltk
import random
import re
import string
```

In [ ]:
```
def remove_special_character(tweet):
    # remove the old style retweet text "RT"
    tweet = re.sub(r'^RT[\s]+', '', tweet)
    # remove hyperlinks
    tweet = re.sub(r'https?:\/\/.*[\r\n]*', '', tweet)
    # remove hashtags
    tweet = re.sub(r'#', '', tweet)
    # remove single numeric terms in the tweet.
    tweet = re.sub(r'[0-9]', '', tweet)
    tweet = re.sub(r'@\w+', '', tweet)
    tweet = re.sub(r'[^a-zA-Z\s]', '', tweet)
    tweet=tweet.lower()
    return tweet
```

executing function

In [ ]:
```
tsla_ml.loc[:, "body"] = tsla_ml['body'].apply(lambda tweet: remove_special
```

In [ ]:
```
aapl_ml.loc[:, "body"] = aapl_ml['body'].apply(lambda tweet: remove_special
```

In [ ]:
```
amzn_ml.loc[:, "body"] = amzn_ml['body'].apply(lambda tweet: remove_special
```

In [ ]:
```
googl_ml.loc[:, "body"] = googl_ml['body'].apply(lambda tweet: remove_speci
```

In [ ]:
```
msft_ml.loc[:, "body"] = msft_ml['body'].apply(lambda tweet: remove_special
```

sentinet analysis

In [ ]:
```
!pip install afinn -q
```

AFINN is a list of words rated with an integer between minus five (negative) and plus five (positive) that is used for sentiment analysis

In [ ]:
```
from afinn import Afinn
afinn = Afinn()
```

In [ ]:
```
tsla_ml['score'] =  tsla_ml['body'].apply(lambda tweet: afinn.score(tweet))
```

In [ ]:
```
aapl_ml['score'] =  aapl_ml['body'].apply(lambda tweet: afinn.score(tweet))
```

In [ ]:
```
amzn_ml['score'] =  amzn_ml['body'].apply(lambda tweet: afinn.score(tweet))
```

In [ ]:
```
msft_ml['score'] =  msft_ml['body'].apply(lambda tweet: afinn.score(tweet))
```

In [ ]:
```
googl_ml['score'] =  googl_ml['body'].apply(lambda tweet: afinn.score(tweet
```

visualsing the tweet scores of each company

In [ ]:
```
rcParams['figure.figsize'] = (4, 2)
tsla_ml.score.plot(kind='hist',range=(-5,5),bins=40,edgecolor='black')
plt.show()
```

In [ ]:
```
aapl_ml.score.plot(kind='hist',range=(-5,5),bins=40,edgecolor='black')
plt.show()
```

In [ ]:
```
amzn_ml.score.plot(kind='hist',range=(-5,5),bins=40,edgecolor='black')
plt.show()
```

In [ ]:
```
googl_ml.score.plot(kind='hist',range=(-5,5),bins=40,edgecolor='black')
plt.show()
```

In [ ]:
```
msft_ml.score.plot(kind='hist',range=(-5,5),bins=40,edgecolor='black')
plt.show()
```

sentiment over time vs stock price

In [ ]:
```
def sentiment_overtime(tweets1,stock,title):
    md2=pd.merge(tweets1, stock, on='date', how='inner')
    visual= md2.groupby('date')['score'].mean().shift(-1).rolling(30).mean(
    fig, ax = plt.subplots()
    ax.plot(visual,color='orange',label='tweets sentiment')
    ax2 = ax.twinx()
    ax2.plot(stock['Close'],label='stock price')
    ax.set_xlabel('year')
    ax.set_ylabel('tweet sentiment')
    ax2.set_ylabel('stock price')
    ax.legend()
    ax2.legend(loc='upper left')
    plt.title(title)
    plt.show()
```

In [ ]:
```
rcParams['figure.figsize'] = (15, 5)
sentiment_overtime(tsla_ml,tsla_stock,'tesla')
```

In [ ]:
```
sentiment_overtime(aapl_ml,aapl_stock,'apple')
```

In [ ]:
```
sentiment_overtime(amzn_ml,amzn_stock,'amazon')
```

In [ ]:
```
sentiment_overtime(googl_ml,googl_stock,'google')
```

In [ ]:
```
sentiment_overtime(msft_ml,msft_stock,'microsoft')
```

there is a relationship between the sentiment of the tweets to the share price of the company

In [ ]: