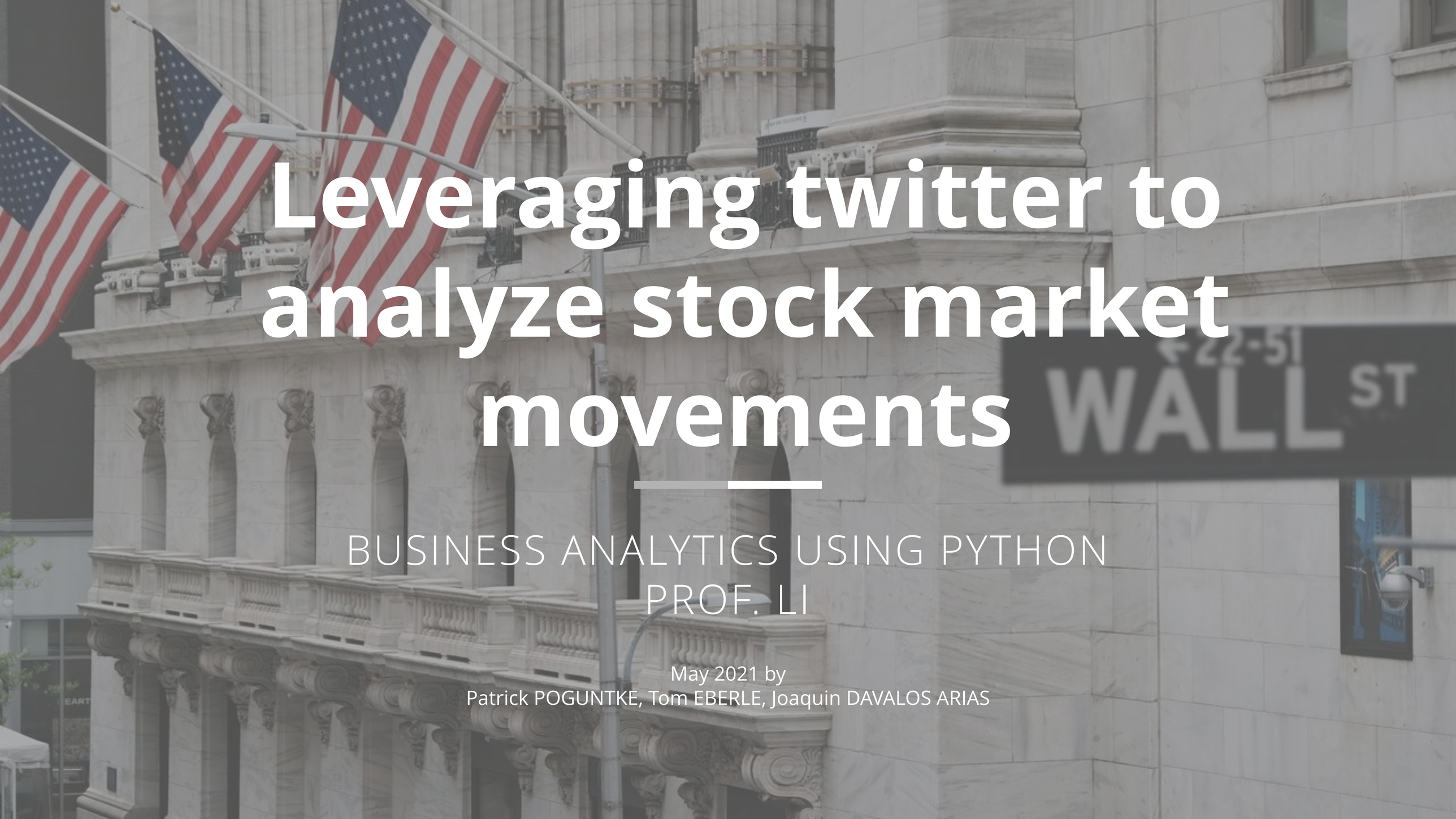


Leveraging twitter to analyze stock market movements

BUSINESS ANALYTICS USING PYTHON
PROF. LI

May 2021 by
Patrick POGUNTKE, Tom EBERLE, Joaquin DAVALOS ARIAS

Agenda

01 The Problem

02 Our Solution

03 Limitations and Improvements

01- The Problem

Innovation to outperform the market

New datasets and speedy reactions can be sources of competitive advantage in the financial markets however half of the market is not ready yet

77%

Of financial institutions think they need to **innovate to survive**

The stock market is a highly **competitive** environment where **hedge funds** and other investors rush to find innovative ways to obtain above average returns.

Innovation pressure

46%

Of managers use **nontraditional data** to decide on investments

Traditional data sources are key but small in scale compared to the **scope of nontraditional data** (private company data, credit card data, social media data). However, this is **not yet exploited** by all.

Scope pressure

50%

Of trading volume is done by **high-frequency traders** (HFT)

Seeing the need to **react quickly** to firm-specific news, macroeconomic **events**, regulatory constraints, **media** endorsements . HFTs) use automated, algorithmic trading, to analyze markets and execute trades in under a millisecond

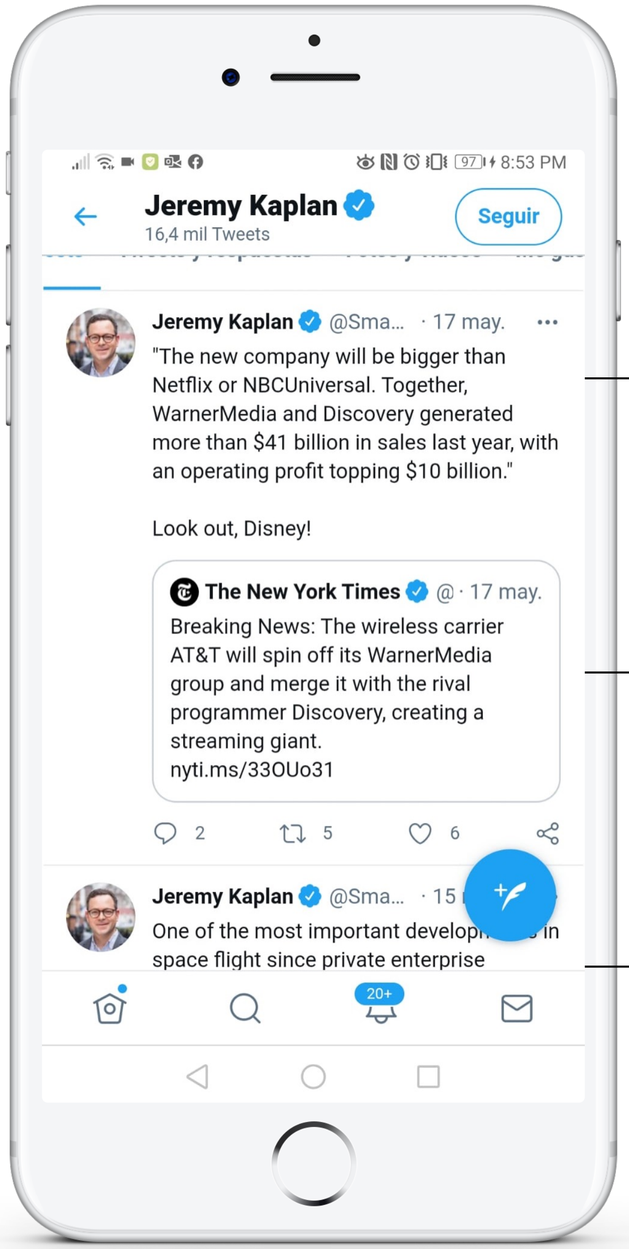
Speed pressure

Sources: (1), (2), (3), (4)

THE PROBLEM

The potential of Twitter

People's quick reaction to news influence the market with positive feedback loops



353M

MONTHLY ACTIVE USERS

→ *HIGH VOLUME ALLOWS FOR INFO TO SPREAD FASTER*

81% OF USERS

USE IT TO READ NEWS AND INTERACT WITH COMPANIES

→ *SOCIAL AND INFORMATIVE USES, PUBLIC DATA*

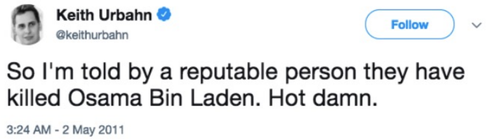
ALL MAJOR

PRESS , MEDIA OUTLETS, AND COMPANIES USE TWITTER

→ *HIGHLY RELEVANT FOR INVESTORS*

A broadcast platform

- Messages are broadcast in real time, developing stories can be updated instantly.
- Ideal platform for those wanting to make announcements, like companies, politicians, users, etc.
- The "retweet" and "follow" functions allow for more closeness and propagation of major announcements.
- Investors create groups tracking all sorts of events, from stock pick decisions to price actions.
- Network effect makes information shared to be more heavily weighted for financial markets.



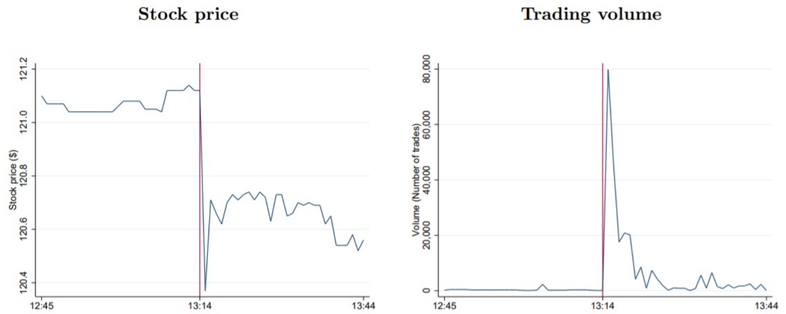
Sources: (5), (6), (7)

Anticipating Herd Behavior

The impact of tweets and threats on the stock market has a major impact on traders and investors: an opportunity to act quickly.

Trump's tweets

- Trump was a heavy users of Twitter as communication channel.
- Since he could enact measures affecting the companies he talked about, the impact on the stock market was direct.
- On 2017, he tweeted: *"Toyota Motor said will build a new plant in Baja, Mexico, to build Corolla cars for U.S. NO WAY! Build plant in U.S. or pay big border tax."*



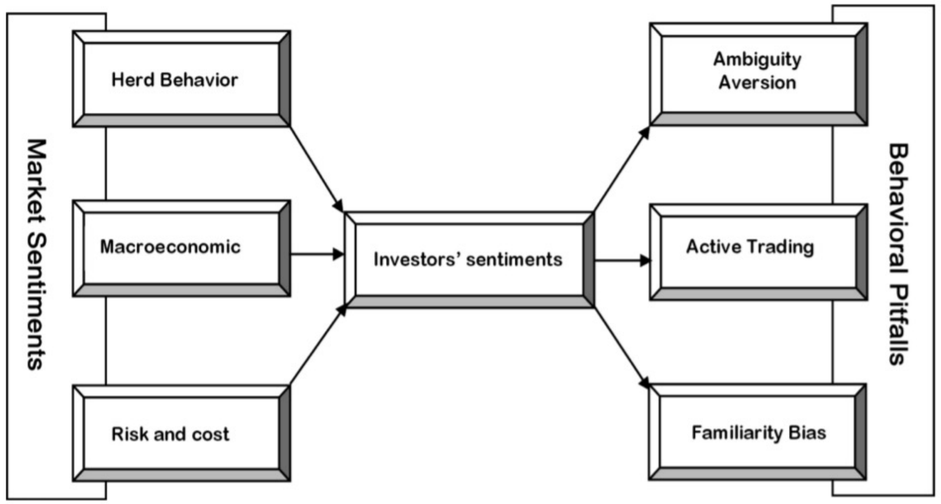
Effect in 60-minute window on Toyota's stock

GameStop case

- A new set of platforms that "democratize" finance (Robinhood).
- In January 2021 Reddit retail investors coordinated to buy stocks, making the share price climb over 1000% in just two weeks.
- By the end of the month short sellers had accumulated losses of more than $5 billion.

Musk's cryptos

- Tesla CEO has sent tweets in the past that affected the performance of cryptocurrencies.
- By "joking" about Doge coin, it made its price increase by 20%.
- Recently he has criticized crypto mining's environmental impact, sending Bitcoin's price down by 15%.



Source: (Haritha & Uchil, 2016)

Herding: Investors following the same information sources and taking similar financial decisions

How to understand the market sentiments from nontraditional sources to take investment decisions?

Sources: (4), (7), (8), (9)

02- Our Solution

Analyzing the social media effect on stock market performance

Our proposed model has 4 functional steps, leveraging web scrapping and sentiment analysis to allow investors take into account twitter reactions

Start

User defines list of companies to analyze and window of time as input

Nasdaq

NYSE

Scrapping tweets from twitter accounts belonging to said companies and their CEOs and from **Reddit investing threads**

A

B

Perform sentiment analysis to detect and label the **level of polarity** (negative, neutral and positive classes)

Scrapping stock prices movements during window of time from financial data sources on the web

C

YAHOO! FINANCE

Investing.com

D

Analyzing the **relationship** between price movements and polarity and the level of **accuracy** for each class

Outcome

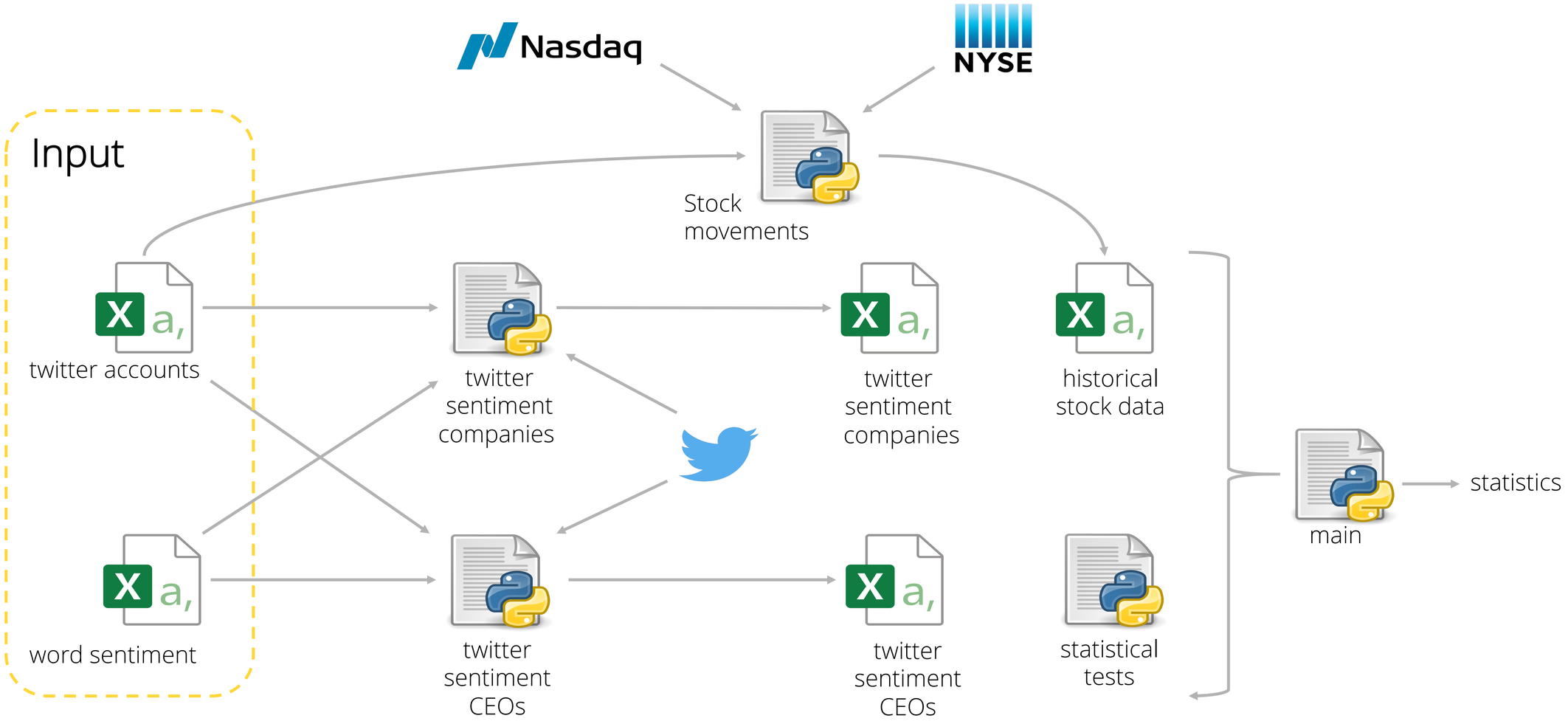
Investors will be able to leverage sentiments of experts to **take better decisions**

Methods

A Snscraper library	B Sentiment analysis labeling
C BS, Yfinance	D Chi Square, Naive Bayes Classifier

Code Structure

The functional steps follow a code structure with a given input



A) Scraping data from Reddit

We scraped Reddit posts and their content from wallstreetbets for a 44-day window and preprocessed the data for analysis. However, identifying companies mentioned in each posts proved to be a challenge.

Data volume

- 10 Reddit pages sample
- 4-day window (16-May to 20-May 2021)
- 100 posts

wallstreetbets

r/wallstreetbets

Joined

CSV

Input format: arguments
- Thread: **Wallstreetbets**
- # pages to scrape

Scrapping library: **BeautifulSoup**
- `soup.find_all` helps to find all posts (sharing same class)
- `post.find().text` helps to further search based on class & attribute
- `soup.find("span", class_="next-button")` helps to find next page

Output (date,_time,title,author,likes,comments,url,content)
preprocessing:
- Processing dates in datetime format
- Data cleaning (e.g., links, special characters)
- Identifying companies mentioned – **very low accuracy (abandoned)**

Wallstreetbets scraping with BeautifulSoup

```
for post in soup.find_all('div', attrs=attrs):
    # Identifying html items to scrape
    title = post.find('p', class_="title").text
    author = post.find('a', class_='author').text
    date = post.find('time')['datetime'][:10]
    _time = post.find('time')['datetime'][11:19]
    comments = post.find('a', class_='comments').text.split()[0]
```

Changing pages

```
# Next page
next_button = soup.find("span", class_="next-button")
next_page_link = next_button.find("a").attrs['href']
time.sleep(2)
page = requests.get(next_page_link, headers=headers)
soup = BeautifulSoup(page.text, 'html.parser')
pageNb += 1
```

Searching for companies mentioned using list of NASDAQ

```
for company in nasdaq:
    # Looking company name
    match_name = title.find(company[1])
    if match_name != -1:
        print(row[0] + '- Found company: ' + str(company))
    # Looking ticker
    match_symbol = title.find('$' + company[0])
    if match_symbol != -1:
        print(row[0] + '- Found ticker: ' + str(company))
```

A) Scraping data from Twitter

We scraped tweets from companies and their CEOs with *snscrape* for a 44-day window and preprocessed the data for analysis

Data volume

- 67 companies (from NASDAQ and NYSE)
  - 66 company twitter accounts
  - 28 CEO twitter accounts
- 44-day window (5-April to 18-May 2021)
- 5,182 tweets
  - 4,474 from companies
  - 708 from CEOs

Input format: CSV containing
- Company Name
- Symbol (e.g., TSLA, GOOG, AAPL)
- Account names

Scrapping library: snscrape
- "from:" specifies twitter accounts to be scraped
- "since:" sets lower bound date limit on query
- "until:" sets upper bound date limit on query

Output (tweet ID, date, time, text, username) preprocessing:
- Deleting replies
- Data cleaning (e.g., links, special characters)
- Adding symbol for ticker (e.g, $APPL)

Tweet scraping with snscrape

```
for index, account in enumerate(account_list):
    symbol = symbol_list[index]
    for tweet in sntwitter.TwitterSearchScraper('from:' + account +
        ' since:2021-04-05 until:2021-05-18').get_items():
        print([tweet.date, tweet.id, tweet.content, tweet.username, symbol])
        tweets_list1.append(
            [tweet.date, tweet.id, tweet.content, tweet.username, symbol])
```

Data cleaning

```
# Dropping replies
tweets_df1['First'] = tweets_df1['Text'].astype(str).str[0]
tweets_df1.drop(tweets_df1[tweets_df1['First'] == '@'].index, inplace=True)
del tweets_df1['First']

# getting rid of everything except letters (a-z)
tweets_df1['Text'] = tweets_df1['Text'].str.lower()
tweets_df1['Text'] = tweets_df1['Text'].apply(lambda x: ' '.join(
    [y for y in x.split() if 'http' not in y]))  # getting rid of the links
tweets_df1['Text'] = tweets_df1['Text'].apply(lambda x: ' '.join(
    [y for y in x.split() if '#' not in y]))  # getting rid of the hashtags
tweets_df1['Text'] = tweets_df1['Text'].replace("'", '', regex=True)
tweets_df1['Text'] = tweets_df1['Text'].replace("’", '', regex=True)
tweets_df1['Text'] = tweets_df1['Text'].replace(
    r'[^a-z ]', ' ', regex=True)
tweets_df1['Text'] = tweets_df1['Text'].str.replace(' +', ' ')
```

Sources: (11), (12)

B) Performing sentiment analysis

We proceeded to measure the sentiment of each tweet, first by defining the dictionary with positive and negative values, and the sentiment measure of each word.

Input: CSV from class material
- word_sentiment.csv
- Some additional negative words added

Sentiment analysis
- Defining word sentiment value
- Splitting sentence into words
- Adding values to return sentence sentiment value

Output (2 data files: Companies and CEOs):
- Tweet ID, date, time, text, username
- Sentiment value

Sentiment analysis

```
SENTIMENT_CSV = "assets/word_sentiment.csv"
NEGATIVE_WORDS = ["not", "dont", "doesnt", "no", "arent", "isnt"]

def word_sentiment(word):
    with open(SENTIMENT_CSV, 'rt', encoding='utf-8') as senti_data:
        sentiment = csv.reader(senti_data)
        for data_row in sentiment:
            if data_row[0] == word.lower():
                sentiment_val = data_row[1]
                return sentiment_val
        return 0

def sentiment(sentence):
    sentiment = 0
    words_list = sentence.split()
    for word in words_list:
        previous_index = words_list.index(word) - 1
        if words_list[previous_index] in NEGATIVE_WORDS:
            sentiment = sentiment + -1 * int(word_sentiment(word))
        else:
            sentiment = sentiment + int(word_sentiment(word))
    return sentiment
```

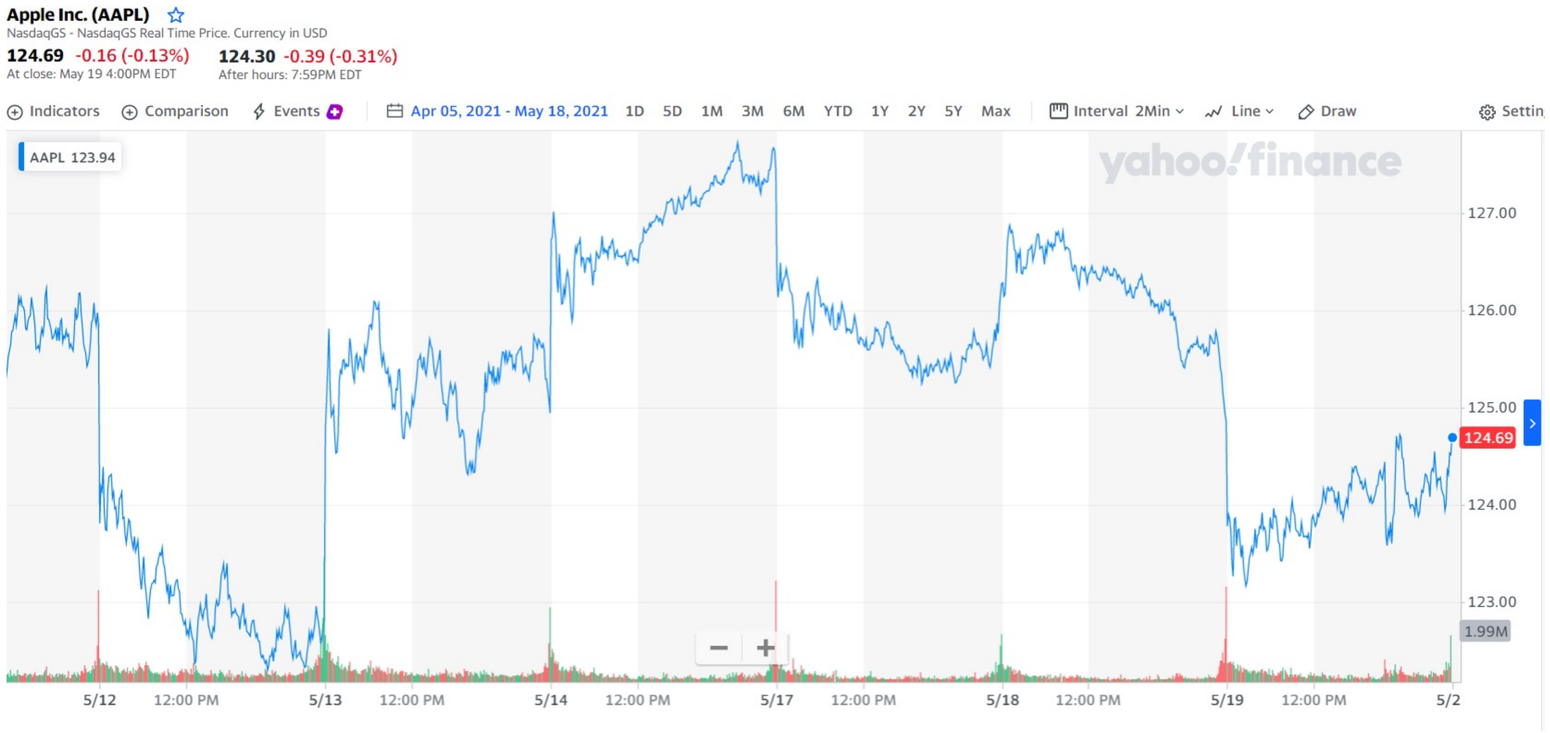
Sources: (13)

C) Obtaining financial data

To complement the data for this analysis we obtained stock price variations of each company for the given time window.

- Stock price movements from **67 companies**
- **44-day** window (5-April to 18-May 2021)
- **2 min** granularity

Stock price movements



Sources: (14)

C) Obtaining financial data

We scraped Yahoo Finance using yfinance, a library specialized in web scrapping for the financial website.

Data volume

- **67 companies** (from NASDAQ and NYSE)
- **44-day** window (5-April to 18-May 2021) with 2min interval
- **400'000+ lines & 52MB of data**

Input format: list of companies symbol
(e.g. ['MSFT', 'AAPL', 'TSLA'])

Scrapping library: **Yfinance**
- yf.download() helps to download stock price data
- yf.Ticker("MSFT").info / financials helps to fetch further financial info
- **Limitation**: max. 7 days at 2' granularity (solve by creating a loop)

Output (Date,Ticker,Adj Close,Close,High,Low,Open,Volume)
preprocessing:
- Processing dates in datetime format
- **Matching Tweets with stock price movement** in interval

Fetching stock price data for each companies

```
    data = yf.download(
        ticker_list, dates[i], dates[i+1], group_by='Ticker', interval="2m")
    # print(data)
    data = data.stack(level=0).rename_axis(
        ['Date', 'Ticker']).reset_index(level=1)
    dataframe = pd.DataFrame(dataframe).append(data)
    print(dataframe)
    time.sleep(2)
```

Creating date loop to go beyond the 7 days limitation

```
dates = pd.date_range(end=datetime.today().strftime(
    '%Y-%m-%d'), periods=8, freq='7D')
print(dates)

i = 0
dataframe = []
for date in dates:
```

Assess price movement for each tweet

```
# Assess price movement
delta = (price_close - price_start) / price_start
print('Delta : ' + str(delta))
print('Start price was ' + str(price_start) +
      ' Close price was ' + str(price_close) + '.'+'Ticker: '+ticker+'. Date: ' + str(date) + '. Delta: ' + str(delta))
if (abs(delta) >= sensitivity):
    # Significant movement
    if delta > 0:
        return 1
    if delta < 0:
        return -1
    else:
        return 0
```

D) Analyzing relationship

With a p-value of 0.022 and 0.096 for Companies and CEOs, respectively, we identified tweet sentiment and stock movement as not independent

Parameters

- Time before tweet: 2 min
- Time after tweet: 4 min
- Sensitivity for significant price movement: 0.0004

Categorical **variables**:
- Tweet sentiment: [positive, negative, neutral]
- Stock movement: [up, down, no movement]

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Chi-squared test
- It tests independence of two categorical variables
- H0: variables are independent
- Cramer's V: measure of the relative strength of an association between two variables (ranges from 0 to 1)

p-value

Coeffients (+ p-values):
- **Companies**: 11.41 (0.022)
- *Significant at 5% level:* sentiment and stock not independent
- Cramer's V Companies: 0.048
- **CEOs**: 7.89 (0.096)
- *Significant at 10% level:* sentiment and stock not independent
- Cramer's V CEOs: 0.11

Chi-squared test

```
# Contingency Table
contingency = pd.crosstab(tweets_df1['Sentiment'], tweets_df1['Movement'])

# Chi-Squared Test
chi2, p, dof, expected = chi2_contingency(contingency)
return chi2, p
```

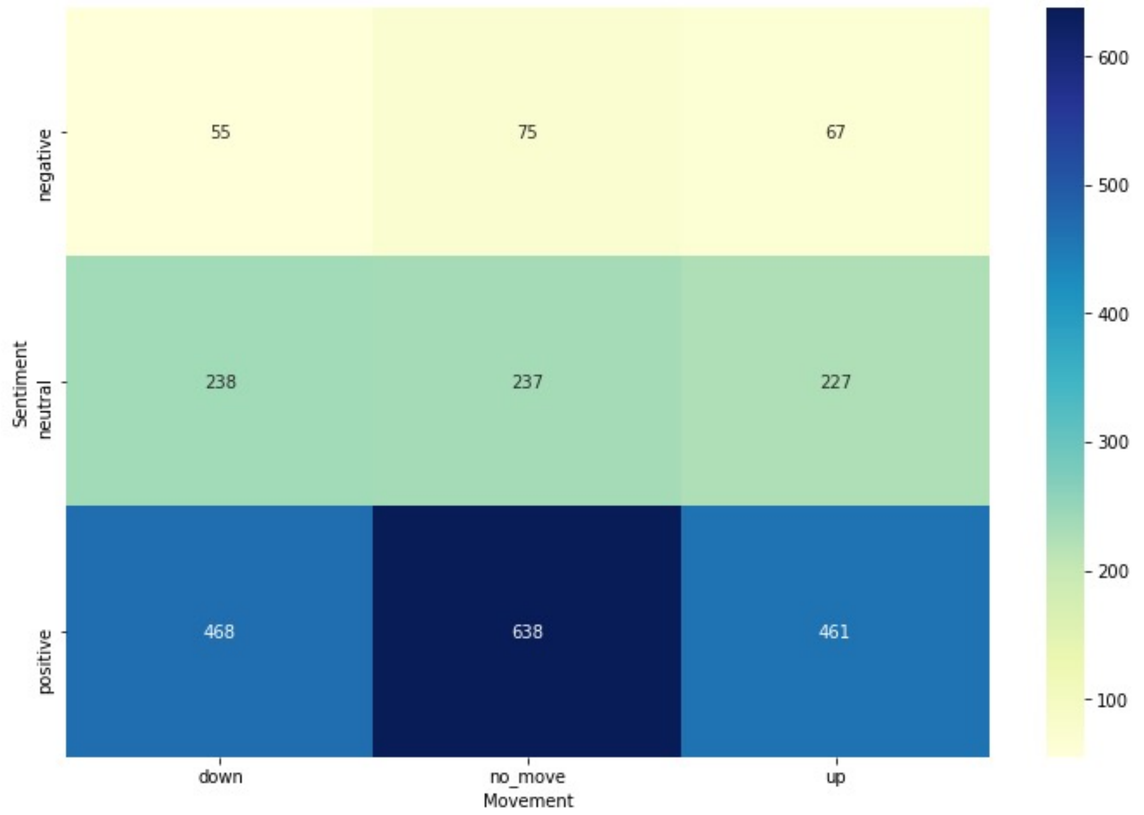
Cramer's V

```
# Cramer's V companies
import numpy as np
data = np.array([[55,75,67], [238,237,227], [468,638,461]])
chi2 = 11.41
n = np.sum(data)
minDim = min(data.shape)-1
V = np.sqrt((chi2/n) / minDim)
print(V)

# Cramer's V CEOs
import numpy as np
data = np.array([[5,2,6], [9,6,19], [77,96,98]])
chi2 = 7.89
n = np.sum(data)
minDim = min(data.shape)-1
V = np.sqrt((chi2/n) / minDim)
print(V)
```

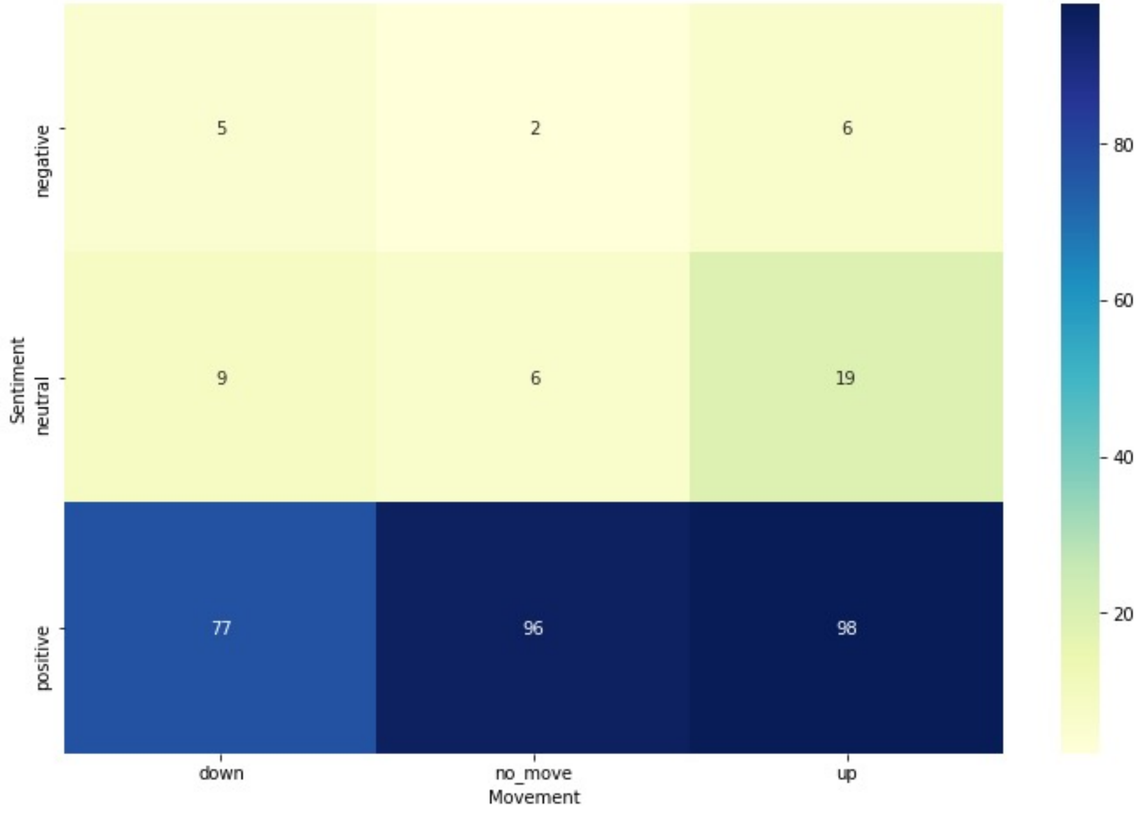
D) Analyzing relationship

Contingency table for companies' tweets



Cramer's V: 0.048

Contingency table for CEOs' tweets



Cramer's V: 0.11

E) Prediction using ML

Using Naïve Bayes to predict stock movements gives no significant results either, with only 45% prediction on CEO's tweets, and 29% on company's tweets.

Prediction capabilities

- CEO tweets: **45%**, companies: **29%**

(x)

Independent **variables**:
- Ticker: e.g. ['AAPL']
- Sentiment: number, e.g. 10
- Volatility: Stock price volatility 120min before (normalized with mean).

Dependent **variables:**
- Stock movement: [positive, negative, none]

Outcome
- Most informative features include **volatility, high amplitude sentiment** and **tickers**.
- Model does not give better results than **randomly guessing**

Example input

```
feature = {'ticker': ticker, 'sentiment': sentiment,
           'volatility': volatility}
feature_label.append(feature)
feature_label.append(stock_mouvement)
# [{'ticker': 'EA', 'sentiment': 8, 'volatility': 0.0}, 1.0],
# [{'ticker': 'EA', 'sentiment': 0, 'volatility': 0.0}, -1.0]
featureset.append(feature_label)
```

Most informative features, Companies' tweets

```
Most Informative Features
              volatility = 0.01             nan : 0.0    =     13.8 : 1.0
                  ticker = 'ACN'            nan : 1.0    =      5.7 : 1.0
                  ticker = 'LULU'           nan : 1.0    =      5.7 : 1.0
                  ticker = 'TSLA'           nan : 1.0    =      5.7 : 1.0
                  ticker = 'F'              nan : 0.0    =      5.3 : 1.0
                  ticker = 'MDLZ'           nan : 0.0    =      5.3 : 1.0
```

Most informative features, CEOs' tweets

```
Most Informative Features
              sentiment = -11               nan : 0.0    =     30.0 : 1.0
              sentiment = -4                nan : 0.0    =     30.0 : 1.0
              sentiment = 14                nan : 0.0    =     30.0 : 1.0
                 ticker = 'DE'              nan : 0.0    =     19.2 : 1.0
                 ticker = 'FDX'             nan : 0.0    =     19.2 : 1.0
```

03- Limitations and Improvements

Roadmap of improvements

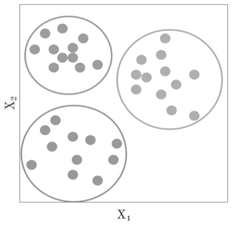
We identified some learning opportunities to make the solution more robust and more responsive with greater scope of analysis

0

Tackle limitations

01

Perform a more sophisticated sentiment analysis with **unsupervised training model** to improve correlation measure



02

Add **more** key nontraditional **data sources** (e.g., Reddit's Wallstreet Bets)

03

Look for **long term trends** in **more indicators**

- Moving averages
- RSI (stocks overbought, oversold)
- VIX (fear, call/put volatility)

04

Identify trending companies in data sources and scrap reactions from **more accounts**, ideally financial influencers or general public

05

Automate the process to be run at certain frequency or with a **trigger** event (e.g., a tweet/ hashtag going viral)

Limitations

- Stock price granularity of 2min – inconsistencies
- Only 44-day window
- Sentiment CSV might not be suitable for stock movements → better with Google Language API
- Company twitter accounts: might only tweet things that are already known →might works better with other accounts
- What is the correct sensitivity? How quickly do stock prices react?
- We don't take magnitude into account in both sentiment and movement
- Majority of tweets positive → maybe use independent twitter accounts (e.g. investment experts) instead of companies

Sources: (10)

Takeaways

Potential of nontraditional public data sets for stock trading

The **sentiments of tweets** have been proven to have **some predictive power for the financial markets**. Maybe this POC raises more questions than it provides answers, but **it contributes** with the development of **sturdier tools** that use alternative data in complex systems such as the financial markets.

Many possible improvements & options for subsequent development

The use of **other types of twitter accounts** (e.g. presidents or other governmental figures, finance experts etc.), **Reddit** and similar **forums**, or more sophisticated **Machine Learning** algorithms and **sentiment analyses** has the potential to result in more viable results.

Sources: (15)

Thank you!

Q&A

May 2021 by
Patrick POGUNTKE, Tom EBERLE, Joaquin DAVALOS ARIAS

References

(1) Thiemann, A. (2020). Innovation and Competition in Financial Markets. https://oecdonthelevel.com/2020/01/14/innovation-and-competition-in-financial-markets/
(2) Lee, M. (2019) How global hedge fund managers can embrace innovation. https://www.ey.com/en_gl/wealth-asset-management/how-will-you-use-innovation-to-illuminate-competitive-advantages
(3) Breckenfelder, J. (2020) Competition among high-frequency traders and market liquidity (2020). https://voxeu.org/article/competition-among-high-frequency-traders-and-market-liquidity
(4) Quaye, I., Mu, Y., Abudu, B., & Agyare, R. (2016). Review of Stock Markets' Reaction to New Events: Evidence from Brexit. Journal of Financial Risk Management, 5, 281-314. http://dx.doi.org/10.4236/jfrm.2016.54025
(5) Twitter Statistics https://backlinko.com/twitter-users
(6) http://cityfalcon.zohosites.com/twitter-tweets-important-for-financial-investors-traders
(7) Tom, A. et al. (2018) Effect of Twitter tweets on the short-term stock prices after Donald Trump presidency. IJRAR
(8) https://www.thetradenews.com/the-reddit-revolt-gamestop-and-the-impact-of-social-media-on-institutional-investors/
(9) https://www.business-standard.com/article/international/dogecoin-surges-on-elon-musk-tweet-as-crypto-rollercoaster-continues-121051401205_1.html
(10) https://www.marketwatch.com/story/use-these-market-indicators-to-predict-stock-moves-2011-02-21
(11) https://betterprogramming.pub/how-to-scrape-tweets-with-snscrape-90124ed006af
(12) https://github.com/JustAnotherArchivist/snscrape
(13) Class material, Python Basics, word_sentiment.csv
(14) https://finance.yahoo.com/
(15) https://blogs.lse.ac.uk/usappblog/2017/10/14/can-twitter-sentiment-predict-stock-market-behaviour/
(16) Valle-Cruz, D., Fernandez-Cortez, V., López-Chau, A. et al. Does Twitter Affect Stock Market Decisions? Financial Sentiment Analysis During Pandemics: A Comparative Study of the H1N1 and the COVID-19 Periods. Cogn Comput (2021). https://doi.org/10.1007/s12559-021-09819-8
(17) Abdulsattar, G. et al. (2017). Stock Market Classification Model Using Sentiment Analysis on
(18) Twitter Based on Hybrid Naive Bayes Classifiers. http://dx.doi.org/10.5539/cis.v11n1p52
(19) Ranco G, Aleksovski D, Caldarelli G, Grčar M, Mozetič I (2015) The Effects of Twitter Sentiment on Stock Price Returns. PLoS ONE 10(9): e0138441. doi:10.1371/journal.pone.0138441
(20) Dhar, S., Bose, I. Emotions in Twitter communication and stock prices of firms: the impact of Covid-19 pandemic. Decision 47, 385–399 (2020). https://doi.org/10.1007/s40622-020-00264-4
(21) Tahir M. Nisar, Man Yeung (2018). Twitter as a tool for forecasting stock market movements: A short-window event study, The Journal of Finance and Data Science, Volume 4, Issue 2, 2018. https://doi.org/10.1016/j.jfds.2017.11.002.
(22) Ge, Qi; Kurov, Alexander; and Wolfe, Marketa Halova (2018). Stock Market Reactions to Presidential Statements: Evidence from CompanySpecific Tweets. Economics Faculty Scholarship. 2. https://creativematter.skidmore.edu/econ_fac_schol/2

Python Tech Stack Used

Libraries:
- yfinance. Yahoo! Finance market data downloader. https://github.com/ranaroussi/yfinance
- Snscrape. A scraper for social networking services (SNS). https://github.com/JustAnotherArchivist/snscrape
- scipy. SciPy: Scientific Library for Python. https://github.com/scipy/scipy
- Pandas. Data structures for data analysis, time series, and statistics. https://github.com/pandas-dev/pandas
- BeautifulSoup
- Requests
- Nltk
- Matplotlib
- Numpy

Collaboration:
- GitHub.
- Google Collab

Editing:
- Visual Studio Code
- Anaconda Spyder
- Google Collab

Full Code Available on Github - https://github.com/tomeberle/Business-analytics