

Analyse Elon's Tweets to automate stock trading

by group 12345

李詳弘 40947026s / 李紀為 40947033S / 郭泰維 40847031S / 何明曦 40947029S / 蕭文政 40947011S

What is the problem?

- Machine Learning & NLP
 - Determine the effect of a tweet to a business
- Automated Trading
 - Less mistakes
- If humans can do it repetitively, computers could do the same

Why is this important?

- Humans have emotions, but computers don't
- Stock trading seems to be a (relatively) monotonous task
- Elon's tweets have significantly affected Tesla's stock
- Learning more about stock trading
- I need money

What to do & How

1. Implement interfaces that interact with ATS
2. Crawl Tweets of Elon Musk ([@elonmusk](#)) in real-time
 1. Also his past tweets for analytical purposes
3. NLP
 1. Running Emotion Recognitions
 2. Determine purposes

Apparently..

Machine Learning
requires some GPU



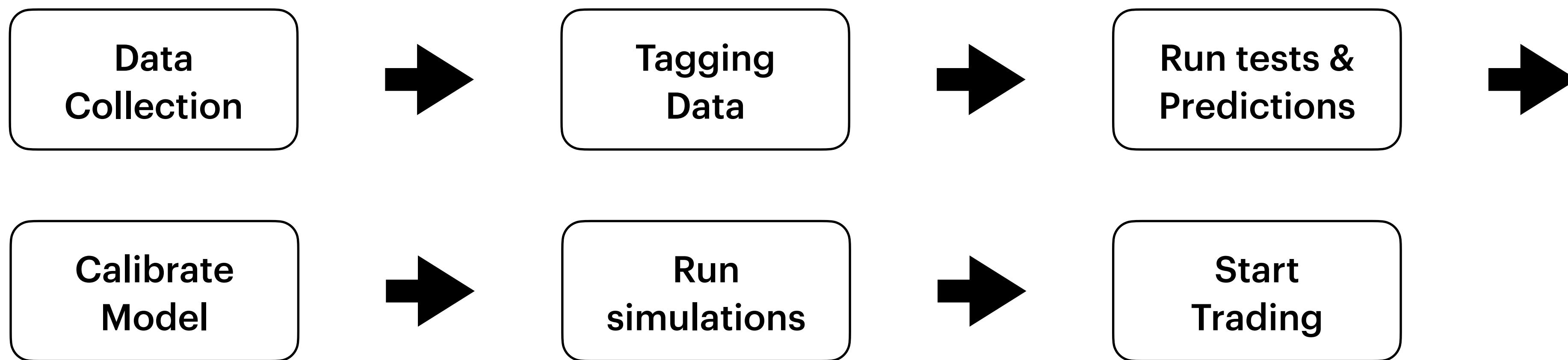
What do we need?

- I'd probably do it on GCP
- Training data already available on Internet

Where do the data come from?

- [https://twitter.com/search?
q=from%3A%40elonmusk%20until%3A2011-12-31](https://twitter.com/search?q=from%3A%40elonmusk%20until%3A2011-12-31)
- Web crawler

Workflow



Our Team

- 李詳弘：設計模型
- 何明曦：收集資料
- 郭泰維：分析股市
- 李紀為：收集資料
- 蕭文政：測試模型

Q & A

Twitter API

developer.twitter.com/en

Twitter Developer Use cases Solutions Products Docs Community Updates Support Developer Portal

Developers

Tap into what's happening.

Publish and analyze Tweets, optimize ads, and create unique customer experiences.

Introducing Early Access to the next generation of the Twitter API. [Learn about v2](#)

Start with a use case

Listen & analyze

Advertise

Publish & curate

Engage

Programmatically

Create connections

developer.twitter.com/en/docs/twitter-api/tweets/full-archive-search/api-reference/get-tweets-search

Twitter Developer Documentation

Documentation

Search the docs

Home > Twitter API

Getting started

Tutorials

Tools and libraries

Migrate

API reference index

The new Twitter API v2

Fundamentals

Tweets

Users

Enterprise Gnip 2.0

Fundamentals

PowerTrack API

Example requests

cURL (default fields) cURL (optional fields)

```
curl "https://api.twitter.com/2/tweets/search/all?query=from%3Atwitterdev&max_results=500&start_time=2020-01-01T00%3A00%3A00Z&end_time=2020-03-31T11%3A59%3A59Z" -H "Authorization: Bearer $BEARER_TOKEN"
```

Example responses

Default Fields Optional Fields

```
1 {  
2   "data": [  
3     {  
4       "id": "1279990139888918528",  
5       "text": "Python now online for you !!\n\nWith the advent and acceptance of AI, Robotics, Python has become an inevitable factor in software development industry and most looked out skill both Nationally and Internationally.\n\nCoupon code: GVUP9\nCall: 9482303905/9482163905 https://t.co/ZFXCDJedAh"  
6     },  
7     {  
8       "id": "1279990133463429120",  
9     }  
10   ]  
11 }
```

Response fields

Name	Type	Description
<code>id</code>	string	Unique identifier of this Tweet. This is returned as a string in order to avoid complications with languages and tools that cannot handle large integers.
<code>DEFAULT</code>		



```
def getTwitterAPI():
    authentication = tweepy.OAuthHandler(consumer_key, consumer_secret)
    authentication.set_access_token(access_token, access_secret)
    return tweepy.API(authentication)
```



```
{  
  "full_text": "RT @SpaceX: Tracking footage from a helicopter of today's  
Falcon 9 launch off LC-39A https://t.co/7rYVZRTS18",  
  "body": "Tracking footage from a helicopter of today's Falcon 9 launch off  
LC-39A"  
}
```

Sentiment Analysis by TextBlob

Sentiment Analysis

The `sentiment` property returns a namedtuple of the form `Sentiment(polarity, subjectivity)`. The polarity score is a float within the range [-1.0, 1.0]. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective.

```
>>> testimonial = TextBlob("Textblob is amazingly simple to use. What great fun!")
>>> testimonial.sentiment
Sentiment(polarity=0.3916666666666666, subjectivity=0.4357142857142857)
>>> testimonial.sentiment.polarity
0.3916666666666666
```



```
def processTweet(tweet):
    return dict(
        text=tweet['full_text'],
        polarity=TextBlob(tweet['body']).sentiment.polarity
    )

results = list(map(processTweet, readJSON(jsonPath)))
```