NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

# Text Analysis using Python

## Sentiment analysis

**Theresa Lee & Nursyafiqah**

Office of Information, Knowledge and Library Services

**April 2024**

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

# Copyright Notice

# Learning Outcomes

At the end of the session, you will be able to:

1. Demonstrate how text cleaning can be done via Python.
2. Perform basic text analysis to explore word count and common words.
3. Perform the task of sentiment analysis through running Python codes.
4. Produce a sentiment analysis chart to analyse the data.

# Text analysis using Python

1. Introduction
2. Setup for hands-on activities
3. Sentiment analysis + hands-on activities
4. Tools and Learning Resources

# Text analysis using Python

1. **Introduction**
2. Setup for hands-on activities
3. Sentiment analysis + hands-on activities
4. Tools and Learning Resources

# Why text analysis using Python?

| | reviewText | Positive |
|---|---|---|
| 0 | This is a one of the best apps acording to a b... | 1 |
| 1 | This is a pretty good version of the game for ... | 1 |
| 2 | this is a really cool game. there are a bunch ... | 1 |
| 3 | This is a silly game and can be frustrating, b... | 1 |
| 4 | This is a terrific game on any pad. Hrs of fun... | 1 |
| ... | ... | ... |
| 19995 | this app is fricken stupid.it froze on the kin... | 0 |
| 19996 | Please add me!!!!! I need neighbors! Ginger101... | 1 |
| 19997 | love it! this game. is awesome. wish it had m... | 1 |
| 19998 | I love love love this app on my side of fashio... | 1 |
| 19999 | This game is a rip off. Here is a list of thin... | 0 |

20000 rows × 2 columns

"love it! This game is awesome..."

"This is a really cool game..."

"This is a silly game and can be frustrating..."

"This game is a rip off..."

# Text analysis using Python

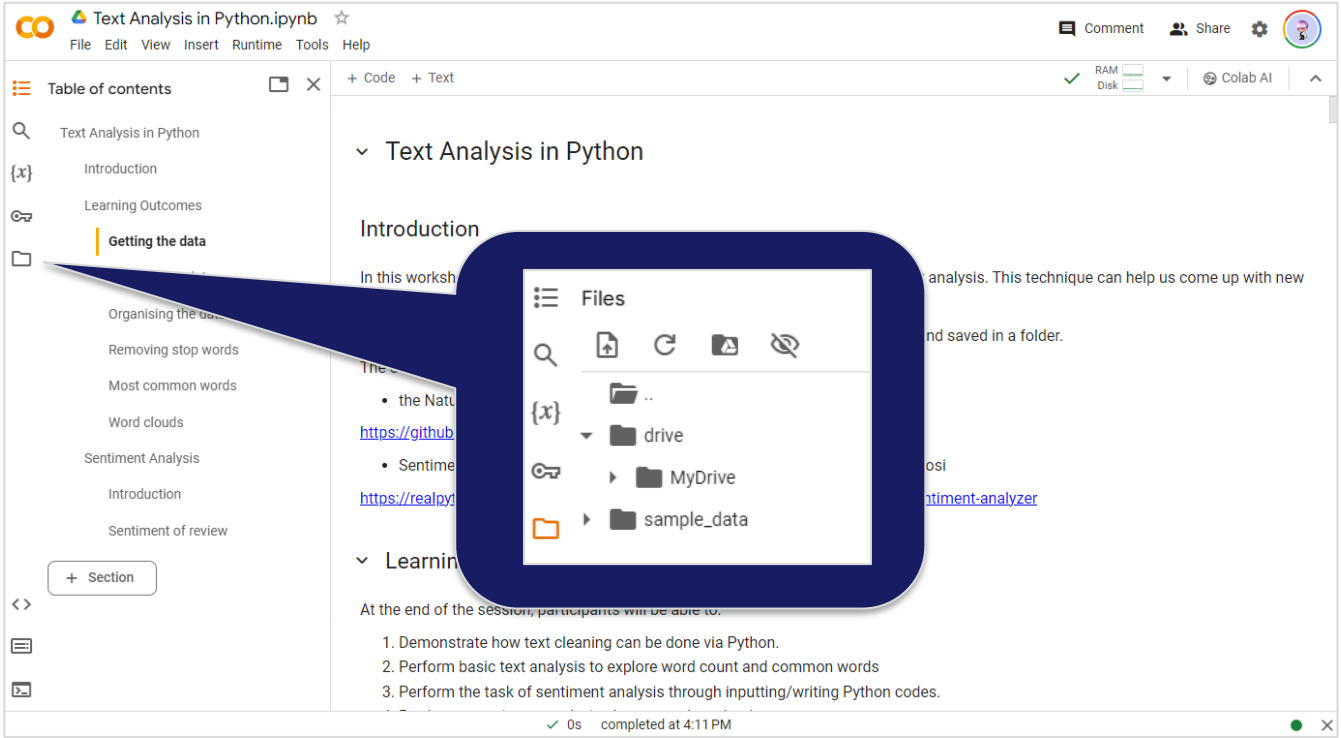
1. Introduction
2. **Setup for hands-on activities**
3. Sentiment analysis + hands-on activities
4. Tools and Learning Resources

# Platforms

Google Colab

Google Drive

# What is Google Colab?

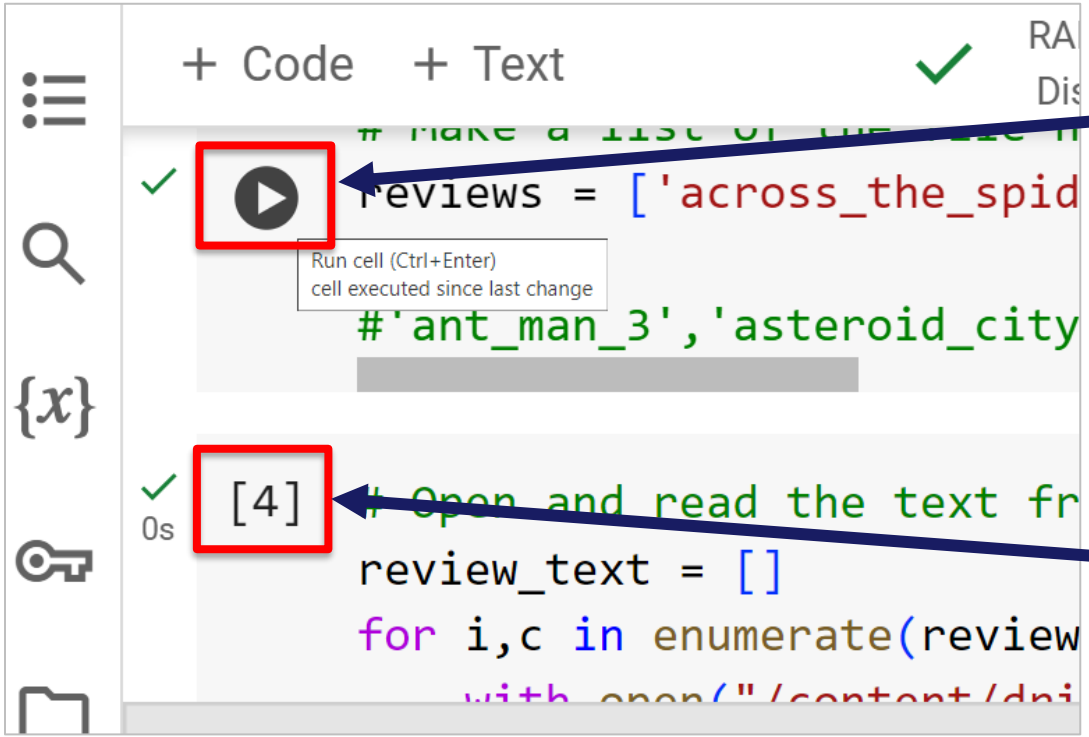


# How it works

```
[ ] ## Cell 1
    # Make a list of the file names of the downloaded movie reviews

    reviews = ['across_the_spiderverse','the_boy_and_the_heron','ant_man_3',
               'asteroid_city','barbie','dungeons_and_dragons', 'elemental',
               'guardians_of_the_galaxy_3','little_mermaid','mission_impossible_7',
               'oppenheimer','past_lives','renfield','saltburn','suzume','talk_to_me',
               'wish','wonka','killers_of_the_flower_moon','john_wick_4']
```

```
[ ] ## Cell 2
    # Open and read the text from the movie review files

    review_text = []
    for i,c in enumerate(reviews):
        with open("/content/drive/MyDrive/Colab Notebooks/movie_reviews/" + c + ".txt", "rb") as file:
          text = file.read()
        review_text.append(text)

    print(review_text[:1])
```

# How it works – cont'd



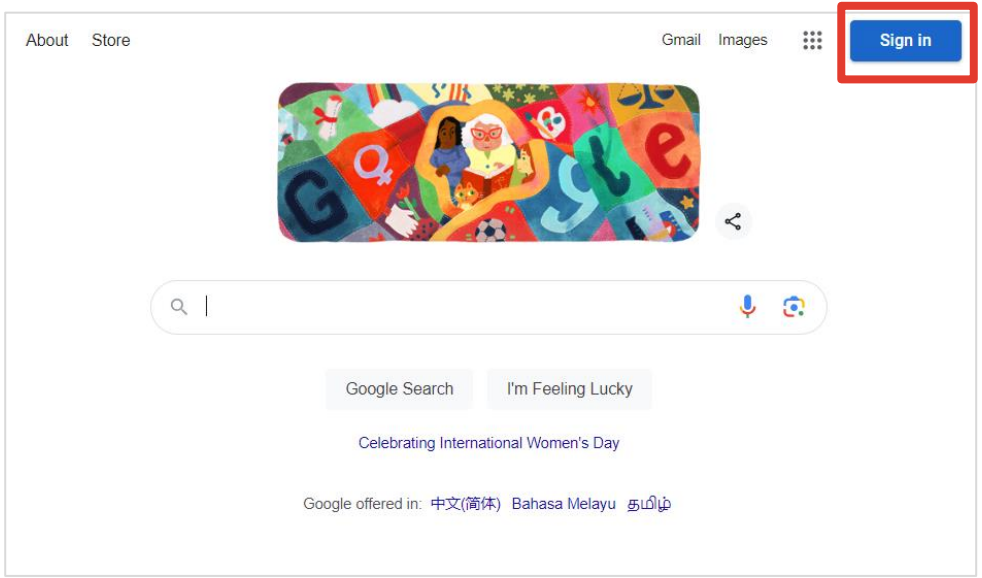

Press the triangle 'Play' button to run the code in the cell

When the cell is finished running, a number in square brackets will appear

# What you need to do

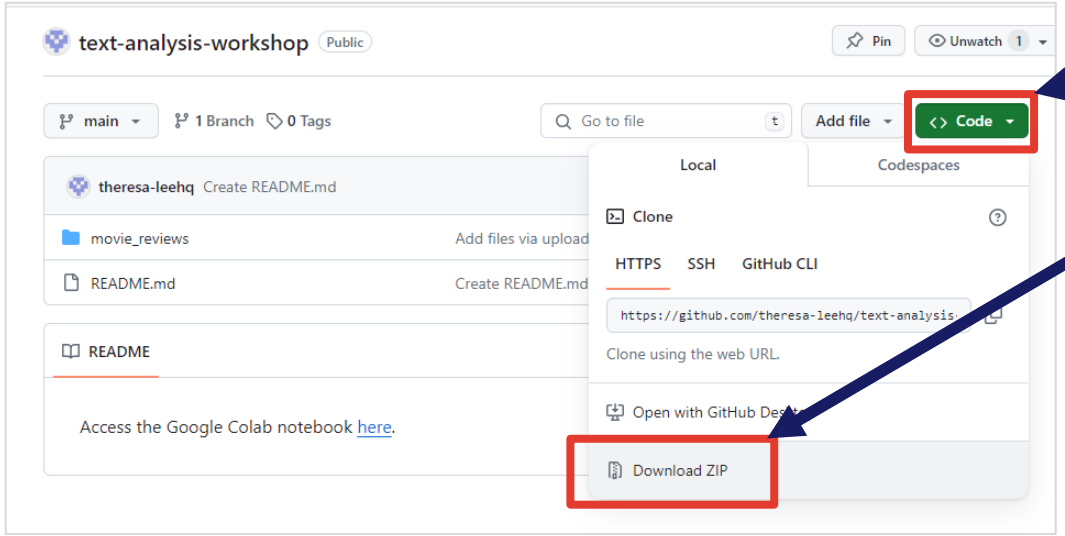
1. Open browser (Chrome or Firefox preferred), go to Google.com and sign in
2. Download lesson materials from GitHub
3. Click on Colab Notebook link on GitHub
4. Duplicate the Colab notebook
5. Grant Colab access to Google Drive
6. Upload “Movie reviews” folder to Colab Notebooks folder on Google Drive

# 1. Open browser (Chrome or Firefox preferred), go to Google.com and sign in (2 mins)



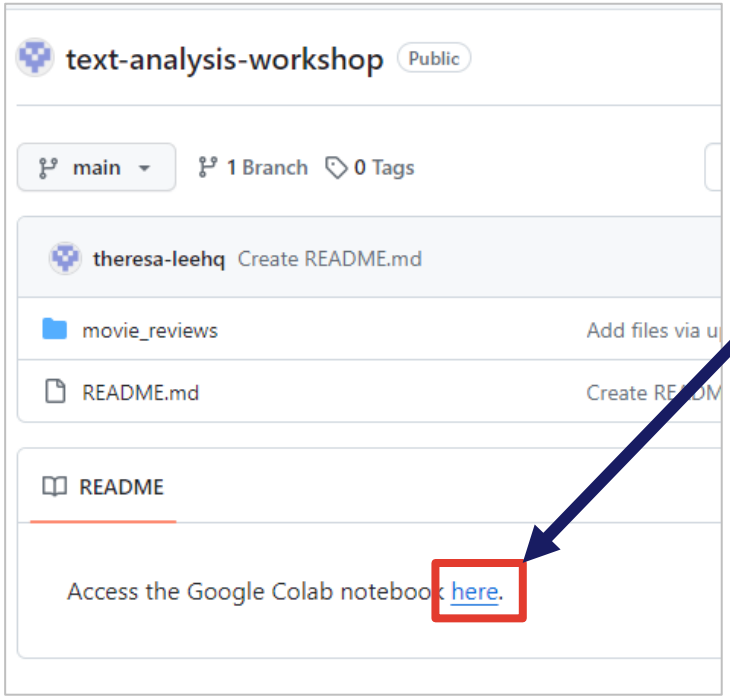
# 2. Download lesson materials (2 mins)

Go to https://bit.ly/ntulib_pyta and download




1. Click the green **'Code'** button
2. Click **'Download ZIP' and save to Desktop.**
3. Unzip the folder.

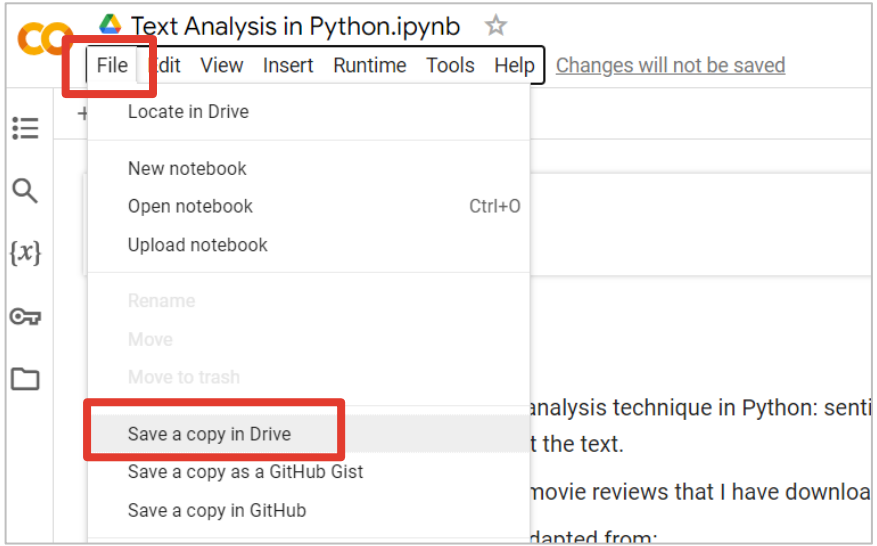
# 3. Click on Colab Notebook link on GitHub (1 min)




Click this link.

It will redirect you to the Colab notebook.

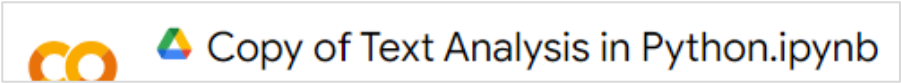
# 4. Duplicate the Colab notebook (1 min)




Click on **File > Save a copy in Drive.**

A new tab will open with a duplicate of the Colab notebook.

The title at the top will show as:

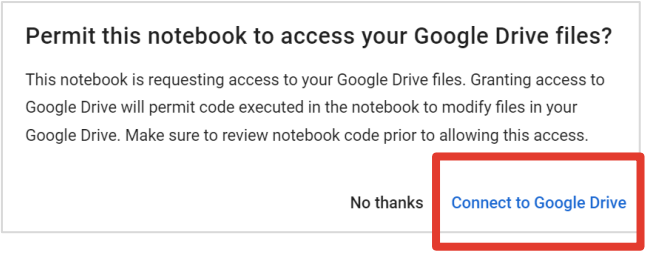


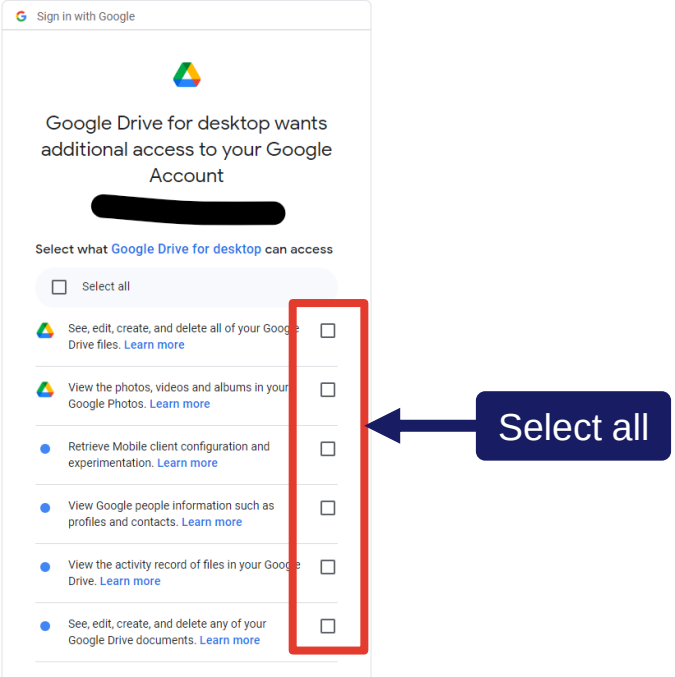
# 5. Grant Colab access to Google Drive (2 mins)

1. Locate and run **Cell 0**.

```
## Cell 0
#Connects this notebook to your Google Drive

from google.colab import drive
drive.mount('/content/drive')
```

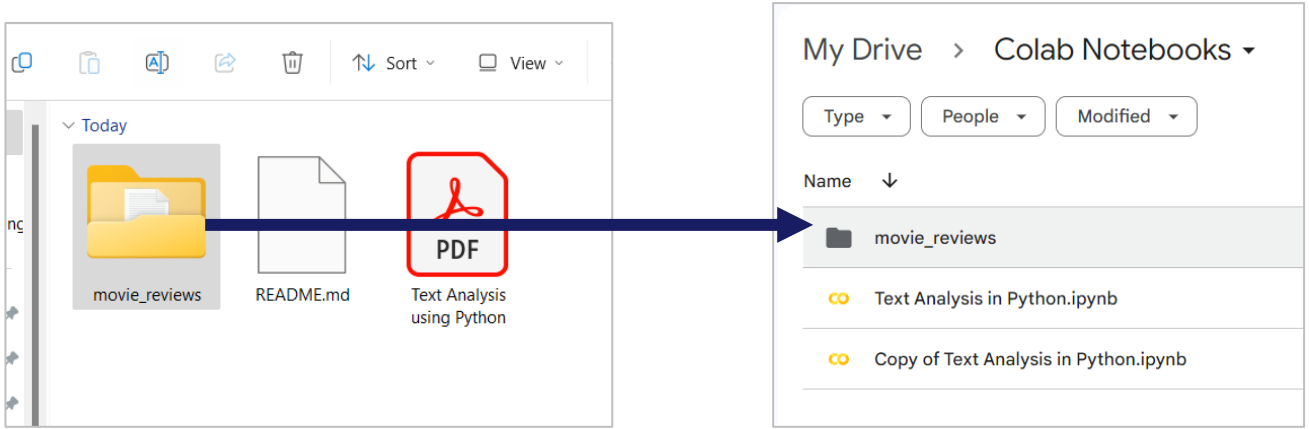
2. When prompted, click **'Connect to Google Drive'**, and **'Select all'.**

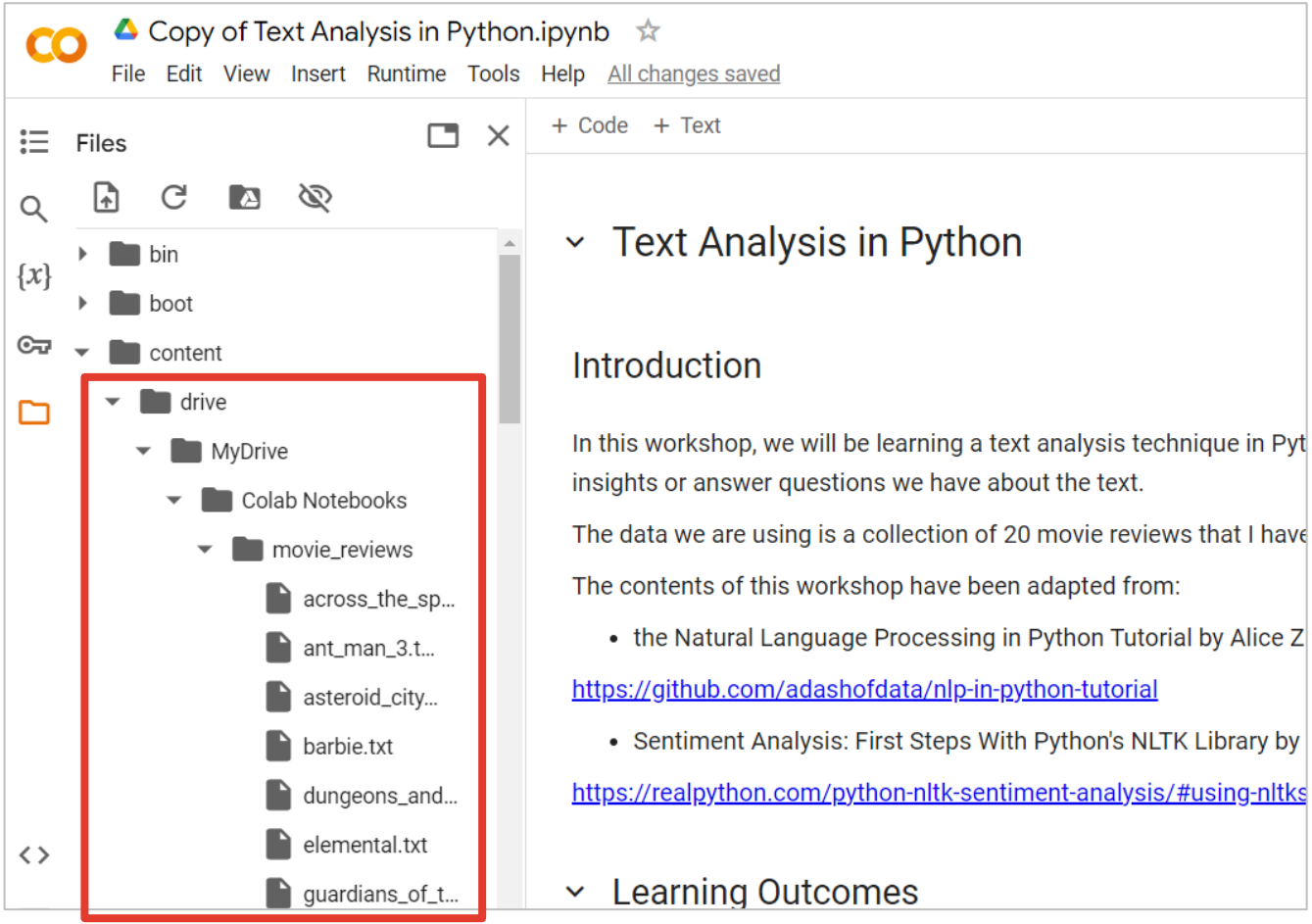


# 6. Upload “Movie reviews” folder to Colab Notebooks folder on Google Drive (2 mins)

1. Go to Google Drive. A ‘**Colab Notebooks**’ folder would have been created automatically.
2. Open the ‘**Colab Notebooks**’ folder.
3. Locate the workshop materials folder on your desktop and drag the **movie_reviews folder** into the **‘Colab Notebooks’** folder.





# All set up?

# Text analysis using Python

1. Introduction
2. Setup for hands-on activities
3. **Sentiment analysis + hands-on activities**
4. Tools and Learning Resources

# Sentiment Analysis

## Activity #1

### Getting, cleaning & organising the data

1. Open text files and load data into Python

2. Clean the data
   - Standardise text and remove irrelevant characters (punctuation, numbers)
   - Remove stop words (e.g. the, a, it, is)
   - Lowercase letters

## Activity #2

### Exploratory data analysis

- Find word frequency
- Create word clouds

## Activity #3

### Perform sentiment analysis

# Sentiment Analysis

## Activity #1

### Getting, cleaning & organising the data

1. Open text files and load data into Python

2. Clean the data
   - Standardise text and remove irrelevant characters (punctuation, numbers)
   - Remove stop words (e.g. the, a, it, is)
   - Lowercase letters

## Activity #2

### Exploratory data analysis

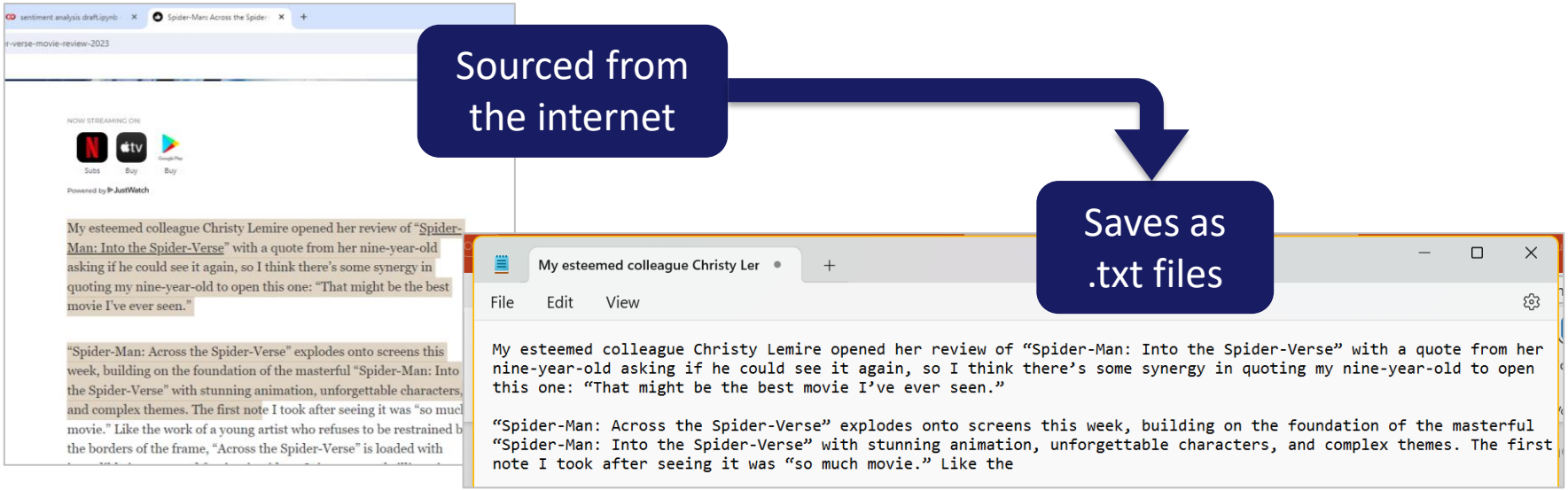
- Find word frequency
- Create word clouds

## Activity #3

### Perform sentiment analysis

# Data used in today's activity

20 movie reviews downloaded from the internet and saved as .txt files



# Final Product of Data Cleaning

Input: Corpus – a collection of texts

| Review name | Text |
|---|---|
| across_the_spiderverse | esteemed colleague christy lemire opened her review of into the with a quote from her nineyearold asking if he could see it again so i think s... |
| ant_man_3 | and the wasp quantumania is an atrocious movie but atrocious in a way that marvel movies rarely are up until now the films of the mcu have for t... |
| asteroid_city | the latest from wes anderson is filled with the assiduous visuals mythic faces and charming curiosities that you expect from this singular filmm... |
| ... | ... |

# Data cleaning

My esteemed colleague Christy Lemire opened her review of “Spider-Man: Into the Spider-Verse” with a quote from her nine-year-old asking if he could see it again, so I think there’s some synergy in quoting my nine-year-old to open this one: “That might be the best movie I’ve ever seen.”

- Remove punctuation
- Remove numbers
- Lowercase letters

my esteemed colleague christy lemire opened her review of spider man into the spider verse with a quote from her nine year old asking if he could see it again so i think there s some synergy in quoting my nine year old to open this one that might be the best movie i ve ever seen

```
[24] # Apply a first round of text cleaning techniques
     import re
     import string

     def clean_text_round1(text):
         '''Make text lowercase, remove text in square brackets, remove punctuation and remove words containing numbers.'''
         text = text.lower()
         text = re.sub('\[.*?\]', '', str(text))
         text = re.sub('[%s]' % re.escape(string.punctuation), '', str(text))
         text = re.sub('\w*\d\w*', '', str(text))
         return text

     round1 = lambda x: clean_text_round1(x)
```

# Data cleaning - Tokenization

Tokenization – split text into smaller pieces (i.e. tokens). Most commonly tokens are words or sentences.

my esteemed colleague christy lemire opened her review of spider man into the spider verse with a quote from her nine year old asking if he could see it again so i think there s some synergy in quoting my nine year old to open this one that might be the best movie i ve ever seen

- Remove stop words

esteemed colleague christy lemire opened review spider man spider verse quote nine year old asking could see i think synergy quoting nine year old open one might best movie ever seen

**Activity #1**
**10 mins**

# Getting, cleaning & organising the data

1. Run Cells 1 to 13
2. Observe the output

# Sentiment Analysis

## Activity #1

### Getting, cleaning & organising the data

1. Open text files and load data into Python
2. Clean the data
   - Standardise text and remove irrelevant characters (punctuation, numbers)
   - Remove stop words (e.g. the, a, it, is)
   - Lowercase letters

## Activity #2

### Exploratory data analysis

- Find word frequency
- Create word clouds

## Activity #3

### Perform sentiment analysis

# Word Clouds

# Word cloud - Before removing stop word

# Find word frequency

Document-term matrix

– Word count of each word in each document

| | abandoned | ability | able | abrupt | abruptly | absencernrnthe | absent | absolute | absolutes | absorbed | ... | youthfriendly | youthful | youtube | youtuber | zaror |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| across_the_spiderverse | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |
| ant_man_3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |
| asteroid_city | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |
| barbie | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |
| dungeons_and_dragons | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |
| elemental | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 |

# Most common words

Find the most common words said in each review

```
across_the_spiderverse
just, miles, film, action, movies, like, superhero, sequence, heroism, gwen, ideas, earns, spot, themes
---
ant_man_3
like, quantum, realm, kang, antman, make, humor, just, marvel, time, onscreen, looks, films, quantumania
---
asteroid_city
like, anderson, town, play, way, film, augie, life, wes, television, looks, makes, films, host
---
barbie
barbie, robbie, world, discover, oscar, moments, just, ken, gosling, gerwig, mattel, role, barbies, movie
---
```

# Remove more stop words

Find which common words can be added to stop words

```
[63] # If more than half of the reviews have it as a top word, exclude it from the list
     add_stop_words = [word for word, count in Counter(words).most_common() if count > 8]
     add_stop_words

     ['like', 'story', 'movie', 'film', 'way', 'just']
```

# Word cloud – After removing stop words

barbie

# Customise the colours

```
## Cell 24

wc = WordCloud(stopwords=stop_words, background_color="white", colormap="Dark2",
               max_font_size=150, random_state=42)
```

## Activity #2
## 10 mins

# Find word frequency + Create word clouds

1. Go to Notebook
2. Run cells 14 to 25

# Sentiment Analysis

## Activity #1

### Getting, cleaning & organising the data

1. Open text files and load data into Python

2. Clean the data
   - Standardise text and remove irrelevant characters (punctuation, numbers)
   - Remove stop words (e.g. the, a, it, is)
   - Lowercase letters

## Activity #2

### Exploratory data analysis

- Find word frequency
- Create word clouds

## Activity #3

**Perform sentiment analysis**

# Introduction to Vader (Cell 26)

| | |
|---|---|
| **neg**<br><br>the negative sentiment score (between 0 to 1) | **pos**<br><br>the positive sentiment score (between 0 to 1) |
| **neu**<br><br>the neutral sentiment score (between 0 to 1) | **compound**<br><br>the overall sentiment score (between -1 to 1) |

```
[ ] ## Cell 26
    #Install and use the VADER module

    import nltk
    nltk.download('vader_lexicon')
    from nltk.sentiment.vader import SentimentIntensityAnalyzer

    sid = SentimentIntensityAnalyzer()

    sid.polarity_scores("I like your dress.")
    #sid.polarity_scores("Your hair looks terrible.")
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
{'neg': 0.0, 'neu': 0.444, 'pos': 0.556, 'compound': 0.3612}
```

# Introduction to TextBlob (Cell 27)

```
[69] #!python -m pip install textblob
     from textblob import TextBlob

     TextBlob("I love movies!").sentiment

     Sentiment(polarity=0.625, subjectivity=0.6)
```

```
[70] TextBlob("This is a terrible film.").sentiment

     Sentiment(polarity=-1.0, subjectivity=1.0)
```

| Sentiment | Polarity |
| --- | --- |
| -1 is negative<br>+1 is positive<br>0 is neutral | -1 is objective<br>+1 is subjective<br>0 is neutral |

**Activity #3**
**Part 1**
**5 mins**

# Introduction to VADER & TextBlob

1. Go to Notebook
2. Run cells 26 & 27

**Activity #3**
**Part 2**
**5 mins**

## Sentiment of Review

1. Go to Notebook
2. Run cells 28 to 32
3. Explanation by instructor

# Conclusion to sentiment analysis

## VADER

- Optimised for social media data
- Better at analysing slang and emojis

## TextBlob

- Can perform other types of analysis in addition to sentiment analysis
- Works better with more formal language and longer text

# Discussion

- Which were the worst reviewed films?
- Did you expect these films to be reviewed so badly?
- Do movie reviews tend to be more subjective or objective?

# End of hands-on activities

# Text analysis using Python

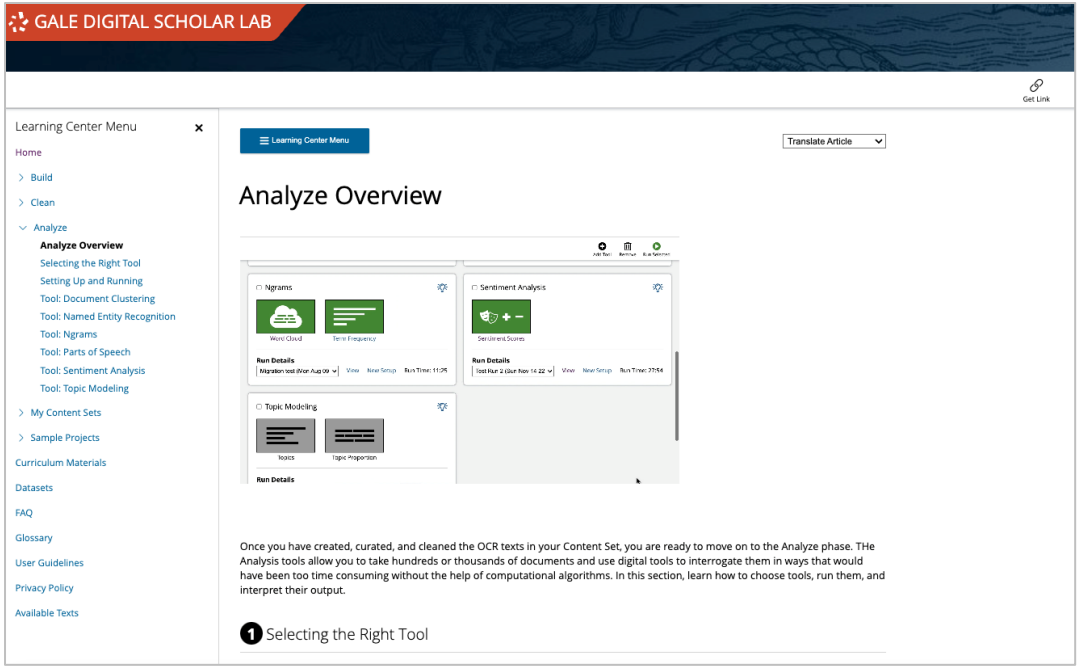
1. Introduction
2. Setup for hands-on activities
3. Sentiment analysis + hands-on activities
4. **Tools and Learning Resources**

# Gale Digital Scholar Lab

- Sentiment analysis
- Topic modelling
- Named entity recognition

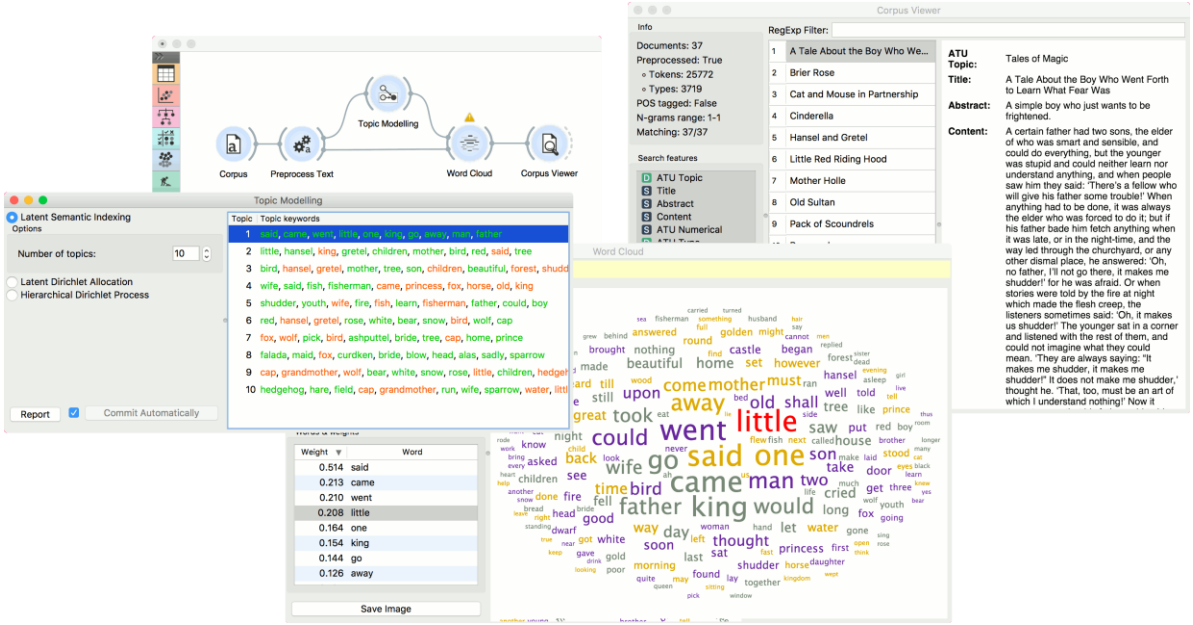
Access through NTU Library Database List:

*https://libguides.ntu.edu.sg/az.php*



# Orange Data Mining

- Text mining
- Sentiment analysis



● Latent Semantic Indexing

Number of topics: 10

○ Latent Dirichlet Allocation

○ Hierarchical Dirichlet Process

☑ Commit Automatically

| Topic | Topic keywords |
|---|---|
| 1 | said, came, went, little, one, king, go, away, man, father |
| 2 | little, hansel, king, gretel, children, mother, bird, red, said, tree |
| 3 | bird, hansel, gretel, mother, tree, son, children, beautiful, forest, shudd |
| 4 | wife, said, fish, fisherman, came, princess, fox, horse, old, king |
| 5 | shudder, youth, wife, fire, fish, learn, fisherman, father, could, boy |
| 6 | red, hansel, gretel, rose, white, bear, snow, bird, wolf, cap |
| 7 | fox, wolf, pick, bird, ashputtel, bride, tree, cap, home, prince |
| 8 | falada, maid, fox, curdken, bride, blow, head, alas, sadly, sparrow |
| 9 | cap, grandmother, wolf, bear, white, snow, rose, little, children, hedgeh |
| 10 | hedgehog, hare, field, cap, grandmother, run, wife, sparrow, water, littl |

| Weight ▼ | Word |
|---|---|
| 0.514 | said |
| 0.213 | came |
| 0.210 | went |
| 0.208 | little |
| 0.164 | one |
| 0.154 | king |
| 0.144 | go |
| 0.126 | away |

Corpus Viewer

Info

Documents: 37
Preprocessed: True
∘ Tokens: 25772
∘ Types: 3719
POS tagged: False
N-grams range: 1-1
Matching: 37/37

Search features

D ATU Topic
S Title
S Abstract
S Content
S ATU Numerical

RegExp Filter:

| | |
|---|---|
| 1 | A Tale About the Boy Who We... |
| 2 | Brier Rose |
| 3 | Cat and Mouse in Partnership |
| 4 | Cinderella |
| 5 | Hansel and Gretel |
| 6 | Little Red Riding Hood |
| 7 | Mother Holle |
| 8 | Old Sultan |
| 9 | Pack of Scoundrels |

**ATU Topic:** Tales of Magic

**Title:** A Tale About the Boy Who Went Forth to Learn What Fear Was

**Abstract:** A simple boy who just wants to be frightened.

**Content:** A certain father had two sons, the elder of who was smart and sensible, and could do everything, but the younger was stupid and could neither learn nor understand anything, and when people saw him they said: 'There's a fellow who will give his father some trouble!' When anything had to be done, it was always the elder who was forced to do it; but if his father bade him fetch anything when it was late, or in the night-time, and the way led through the churchyard, or any other dismal place, he answered: 'Oh, no father, I'll not go there, it makes me shudder!' for he was afraid. Or when stories were told by the fire at night which made the flesh creep, the listeners sometimes said: 'Oh, it makes us shudder!' The younger sat in a corner and listened with the rest of them, and could not imagine what they could mean. 'They are always saying: "It makes me shudder, it makes me shudder!" It does not make me shudder,' thought he. 'That, too, must be an art of which I understand nothing!' Now it

# Microsoft Excel

- Azure Machine Learning add-in

| | A | B | C |
|---|---|---|---|
| 1 | **tweet_text** | Sentiment | Score |
| 2 | Hello, I'm having trouble working this one. In col | positive | 98.800% |
| 3 | OK, so it's Monday morning and I obviously cann | neutral | 51.651% |
| 4 | Hi all, There has been a post previous to this reg | negative | 0.000% |
| 5 | hi, would like to have a formula or vb code for th | positive | 97.175% |
| 6 | Hi,  I need to collect data (selected range) from | positive | 95.626% |

# Why Python for sentiment analysis?

## No-code tools

- Black box
- Limited to functions provided by creators

## Python

- Transparent
- Reproducible
- Customisable

# Limitations

## Limitations of sentiment analysis

- Sarcasm
- Multiple polarity (sentences with ‘but’)
- Change in sentiment over time

## Limitations of AI (machine learning)

- Lack of training data/bad training data
- Difficult to interpret decisions
- Overfitting

# FAQs

1. Why remove punctuations, upper case letters?
2. What are the real-world use cases?

**Any other questions?**

# Feedback Form

https://survey.ntu.edu.sg/efm/se/705E3F172A6B7B2B

# Post class activity (optional)

1. Find a new movie review
2. Pass data through sentiment analyses