





# Exploration of Netizen Concern About Covid-19

**Intro to NLP and Text Mining  
Data Science 4 - Dataloper**

Abiyyu Fathin Derian  
Alifia C. Harmadi  
Dhea Fajriati Anas  
Hendri Prabowo  
Nikolas Rakryan Widagdo

# Table of Contents

01	Introduction and Objectives	
02	Text preprocessing	
03	Feature extraction with Bag of words and TF-IDF technique	
04	Exploration on POS-Tagging and NER	

# Introduction

## Source of Data

<https://twitter.com/>



## Keyword

Covid



## Time of Crawling Data

21/07/2021 - 22/07/2021

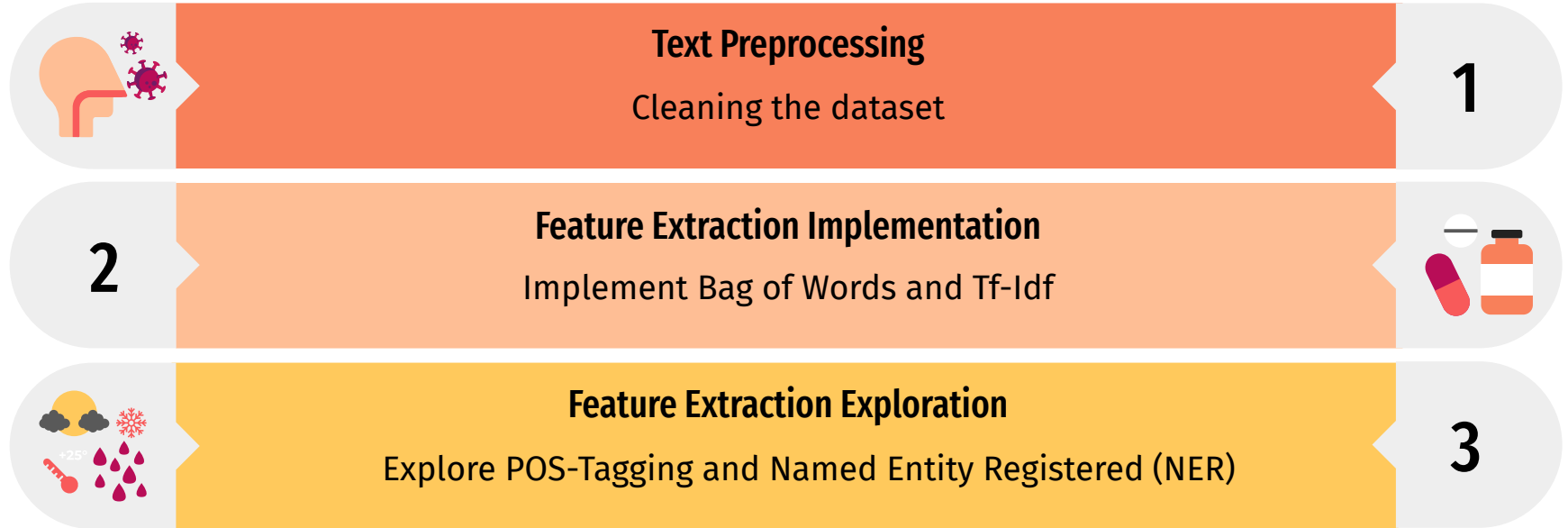


## Total Data

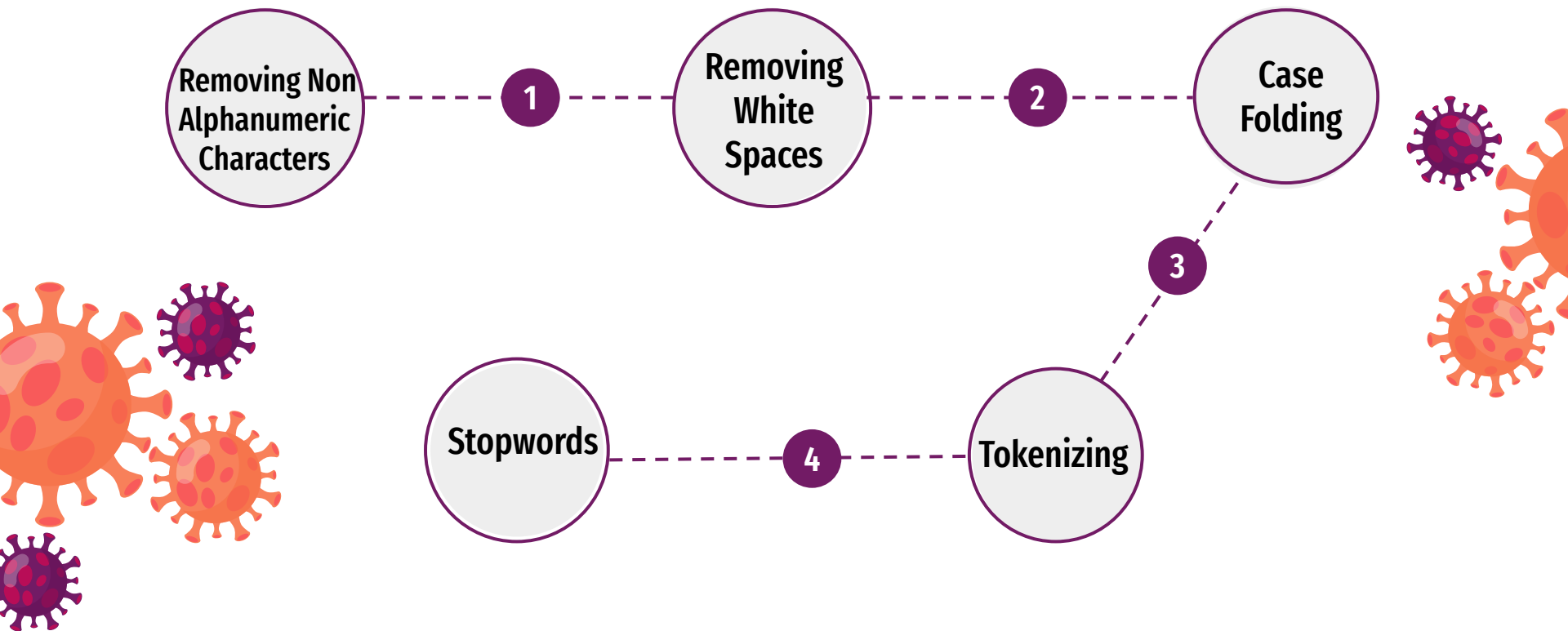
1000 data with 3 columns, i.e. Date,  
Tweet, and User



# Objectives



# Text Preprocessing



# Text Preprocessing

```
def clean_data(text):
    text = re.sub('@([a-zA-Z0-9_]+)', '', text) #menghapus @mention
    text = re.sub('@[^\s]+', '', text)
    text = re.sub('#[\s]+', '', text) #menghapus hashtag
    text = re.sub('RT[\s]+', '', text) #menghapus RT
    text = re.sub('https?:\/\/\S+', '', text) #menghapus hyperlink
    text = re.sub('\d+', '', text) #menghapus angka
    text = re.sub('[^\w\s]', '', text) #menghapus tanda baca
    text = re.sub(r'\b[a-zA-Z]\b', '', text) #menghapus single character
    text = re.sub('\n', '', text) #menghapus \n
    text = re.sub('\r', '', text) #menghapus \r
    text = text.strip() #menghapus whitespace
    text = text.lower() #lowercase
    return text
```

## Before

```
0 @jokowi Kami sekeluarga dari awal covid sdh pr...
1 Pemerintah 'Nunggak' Bayar Klaim Covid-19 ke R...
2 @CTNurza @DoktorSamhan Masalahnya manusia yg t...
3 Ketawa saja bung @Dennysiregar7 , mereka itu o...
4 Aku iki cuma overthinking ae. Gumun juga klo n...
...
995 @CNNIndonesia Dalam keadaan darurat, prosesnya...
996 @zouloutchaaaing rohi diri lvaccin hari denya ...
997 TNI-Polri bagikan Masker kepada masyarakat gun...
998 Hallo Sobat Polri... anak-anak sangat rentan t...
999 Bupati Karawang Cellica Nurrachadiana Kembali ...
Name: Tweet, Length: 1000, dtype: object
```

## After

```
0 kami sekeluarga dari awal covid sdh proses pak...
1 pemerintah nunggak bayar klaim covid19 ke rs r...
2 masalahnya manusia yg tak berakal tidak berota...
3 ketawa saja bung mereka itu orangorang yg ku...
4 aku iki cuma overthinking ae gumun juga klo nd...
...
995 dalam keadaan darurat prosesnya jangan lama2 b...
996 rohi diri lvaccin hari denya kaml bl covid
997 tnipolri bagikan masker kepada masyarakat guna...
998 hallo sobat polri anakanak sangat rentan terha...
999 bupati karawang cellica nurrachadiana kembali ...
Name: Tweet, Length: 1000, dtype: object
```



# Text Preprocessing

## Stopword

	slang	formal
0	wowww	wow
1	aminn	amin
2	met	selamat
3	netaas	menetas
4	keberpa	keberapa
...	...	...
15001	gataunya	enggak taunya
15002	gtau	enggak tau
15003	gatau	enggak tau
15004	fans2	fan-fan
15005	gaharus	enggak harus

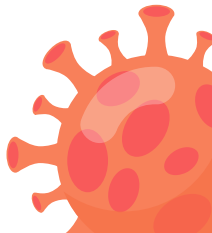
Source:

<https://github.com/nasalsabila/kamus-alay/blob/master/colloquial-indonesian-lexicon.csv>

```
indo = stopwords.words('indonesian')
indo.extend(['yg', 'nya', 'dgn', 'dg', 'dr', 'ya', 'yaa',
            'aja', 'utk', 'ni', 'tp', 'amp', 'dah', 'krn',
            'udah'])
```

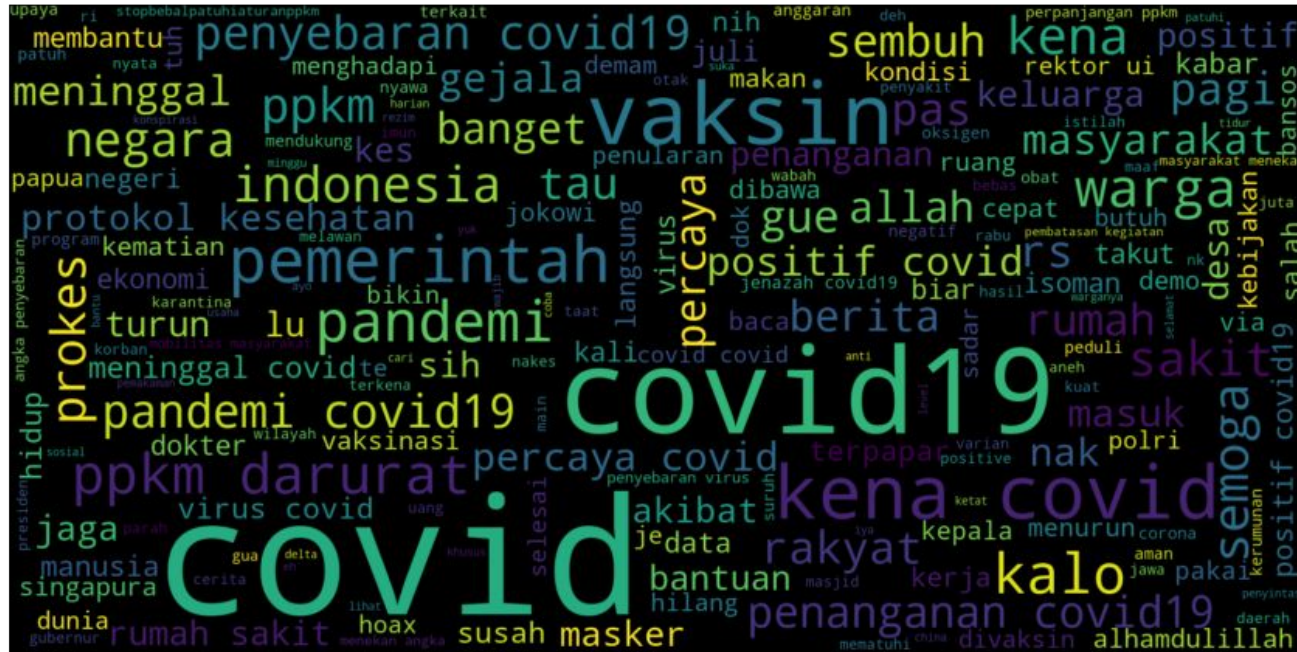
Source:

- Nltk.corpus
- Add manually





# The Most Frequent Word



## TOP 10

```
('covid', 661),
('covid19', 347),
('kena', 109),
('ppkm', 101),
('vaksin', 100),
('pandemi', 72),
('pemerintah', 60),
('positif', 54),
('darurat', 53),
('masyarakat', 52),
```



# Feature Extraction

## Bag of Words

Using library sklearn

---

## TF-IDF

Using library sklearn



## POS-Tagging

Using library flair

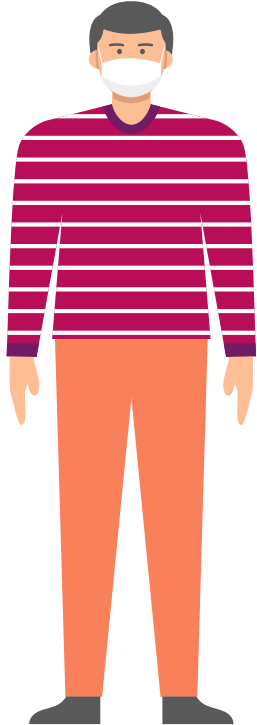
---

## NER

Using library SpaCy



# Bag of Words (CountVectorizer)



## Import Library

Get the library from  
`sklearn.feature_extraction.text`

1

## Word Extraction

Extract the single word from sentence

2

## Word Occurrence

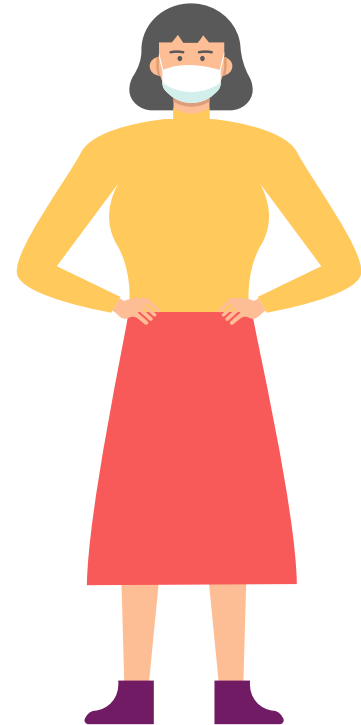
Count the word occurrence in each sentence

3

## DataFrame

Put the result into a DataFrame

4

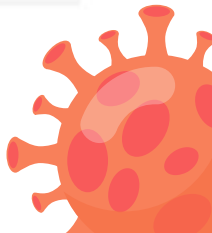




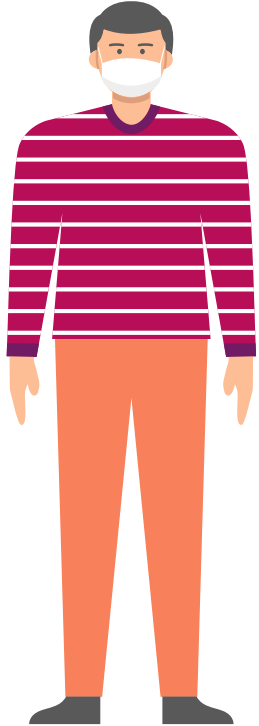
# Bag of Words (CountVectorizer )

	00	000	000t	014	0500	0526	066	072021	0730	10	...	yustisi	yusuf	yuuuuu	zaman	zayed	zodiak	zona	zonasi	zubaidah	๕๓
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

1000 rows x 4899 columns



# TF-IDF



## Import Library

Get the library from  
`sklearn.feature_extraction.text`

1

## Word Extraction

Extract the single word from sentence

2

## Word Occurrence Ratio

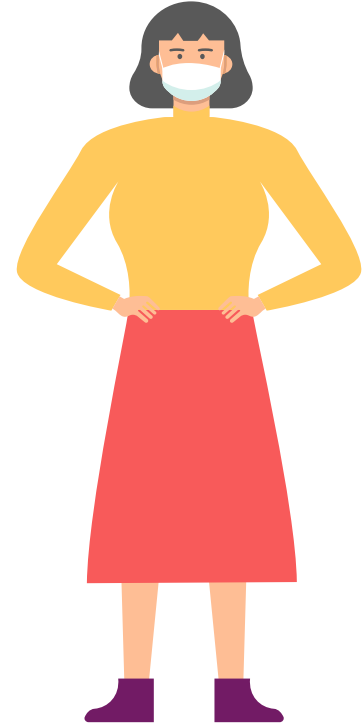
Count the word occurrence ratio in each sentence and whole text

3

## DataFrame

Put the result into a DataFrame

4

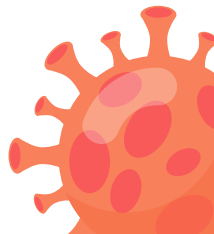




# TF-IDF

	00	000	000t	014	0500	0526	066	072021	0730	10	...	yustisi	yusuf	yuuuuu	zaman	zayed	zodiak	zona	zonasi	zubaidah	zē
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

1000 rows x 4899 columns



# POS (Part Of Speech) Tagging

Classify the word into their part of speech based on its definition and its context such as Subject, Noun, Object, Verb, so on.

Two main parts in POS Tagging Classification Technique

Tagger

a method that labels words as one of several categories to identify the word's function in a given language

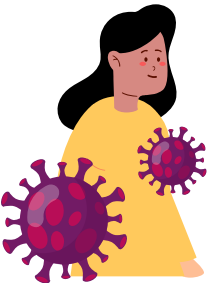
Corpus

A collection of words that already tagged

Two Types of POS Tagger:

Rule-Based POS Tag

Stochastic Tagger



# POS (Part Of Speech) Tagging Experiment



## Import Library

Import Flair library and create  
Tagger and Corpus

## Train Model

Train Tagger and Corpus in  
the Flair Model

## Apply Model

Apply the model to  
preprocessed data

## Input:

```
from flair.data import Sentence
sentence = Sentence('arahan menteri desa dalam penanganan covid')
tag_pos = SequenceTagger.load('resources/taggers/example-universal-pos/best-model.pt')
tag_pos.predict(sentence)
```

## Output:

```
arahan <NOUN> menteri <NOUN> desa <NOUN> dalam <ADP> penanganan <NOUN> covid <PUNCT>
```



# Named Entity Recognition



01

Task of identifying and categorizing key information (entities) in text



02

Classify words into its predefined categories, such as person, organizations, locations, expressions of times, quantities, percentages, etc.



03

NER can be used to retrieve data and information faster from a wide variety of textual information.



04

Experimentation with NER using Spacy

# Named Entity Recognition

1

## Import Library

Get the library from  
spacy

2

## Train Model

Train the model using  
an existing dataset  
(ner\_spacy\_fmt\_data  
sets.pickle)

3

## Apply Model

Apply the model to  
preprocessed data

## Input:

```
doc = nlp("arahan menteri Desa dalam penanganan Covid")
print(doc.ents)
print("Entities", [(ent.text, ent.label_) for ent in doc.ents])
```

## Result:

```
(Desa,)
Entities [('Desa', 'ORGANIZATION')]
```

# Thank you