COVID-19 Twitter Dataset: In-Depth Analysis of Text Clustering and Sentiment Prediction

Team 10
112598017 楊淨雯
112598028 余珮綺

Outline

01
Introduction

02
About Data Set

03
EDA on each columns

04
Text Clustering using k-means and PCA

05
Sentiment Prediction

06
Conclusion

01

Introduction

Introduction

- Following the declaration of the COVID-19 pandemic by the World Health Organization (WHO), social media platforms swiftly emerged as the primary channels for people to communicate, express opinions, and obtain the latest information.

- **Goal**
  To comprehensively understand the dynamics and trends on social media during the COVID-19 pandemic through extensive data analysis and exploration.

Introduction

- Data set: **Covid-19 Twitter Dataset** https://www.kaggle.com/datasets/arunavakrchakraborty/covid19-twitter-dataset/data
- The COVID-19 Twitter dataset, covering the period from April 2020 to June 2021.
- The dataset spans multiple phases of the pandemic and encompasses English tweets from around the globe.

02

About Data Set

About Data Set

This dataset has a total of **10** columns

ID	Tweet ID
created_at	Creation Date & Time
original_text	Original Tweet
favorite_count	Favorite Count of tweets
retweet_count	Retweet Count of tweets
hashtags	Hashtags from tweets
user_mentions	User Mentions of tweets
place	Place of tweets
clean_tweet	Clean and Pre-processed tweets
sentiment	Sentiment class

About Data Set

- **Data Pre-Processing**
  Processed data using NLTK-based function: Lowercasing, removing extraneous elements, converting 'covid' to 'covid19', and stemming.

- **Sentiment Analysis**
  Utilized NLTK Sentiment Analyzer for polarity scores—categorized tweets as Positive, Negative, or Neutral based on compound sentiment scores.

03

EDA on each columns

Exploratory Data Analysis, EDA

When Data Preprocessing

NLTK

```
import nltk
nltk.download('stopwords')

from nltk.corpus import stopwords
stopwords = stopwords.words('english')
stopwords.append(" ")
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

NLTK can be used to process human language. In this project we will use stopwords to deal with some word does not have much meaning, such as: the, a, in, ...etc.

```
broadcast_stopwords = spark.sparkContext.broadcast(stopwords)

def remove_tags(text):
    TAG_RE = re.compile(r'<[^>]+>')
    return TAG_RE.sub('', text)

def preprocess_text(sen):
    # Check if sen is None
    if sen is None:
        return ""

    sentence = sen.lower()

    # Remove HTML tags
    sentence = remove_tags(sentence)

    # Remove punctuation and numbers
    sentence = re.sub('[^a-zA-Z]', ' ', sentence)

    # Single character removal
    sentence = re.sub(r"\s+[a-zA-Z]\s+", ' ', sentence)

    # Remove multiple spaces
    sentence = re.sub(r'\s+', ' ', sentence)

    # Split sentence into words
    words = sentence.split()

    # Remove stopwords using the broadcasted list
    filtered_words = [word for word in words if word not in broadcast_stopwords.value]

    # Join words back into a sentence with spaces
    result_sentence = ' '.join(filtered_words)

    return result_sentence
```

When Data Processing

PySpark

PySpark is the Python API for Apache Spark, a fast and general-purpose big data processing framework.

In this project, we use PySpark to handle our data.

```
from pyspark.sql import SparkSession
from functools import reduce
# init SparkSession
spark = SparkSession.builder.appName("BDM Final").getOrCreate()

# read the csv file
base_path = "/content/drive/MyDrive/巨量資料/BDM_Final/archive-2/"

file_list = [
    "Covid-19 Twitter Dataset (Apr-Jun 2020).csv",
    "Covid-19 Twitter Dataset (Apr-Jun 2021).csv",
    "Covid-19 Twitter Dataset (Aug-Sep 2020).csv"
]

data_frames = []


for file_name in file_list:
    file_path = base_path + file_name
    df = spark.read.csv(file_path, header=True, inferSchema=True)
    data_frames.append(df)

combined_df = reduce(lambda df1, df2: df1.union(df2), data_frames)
```

Used Data

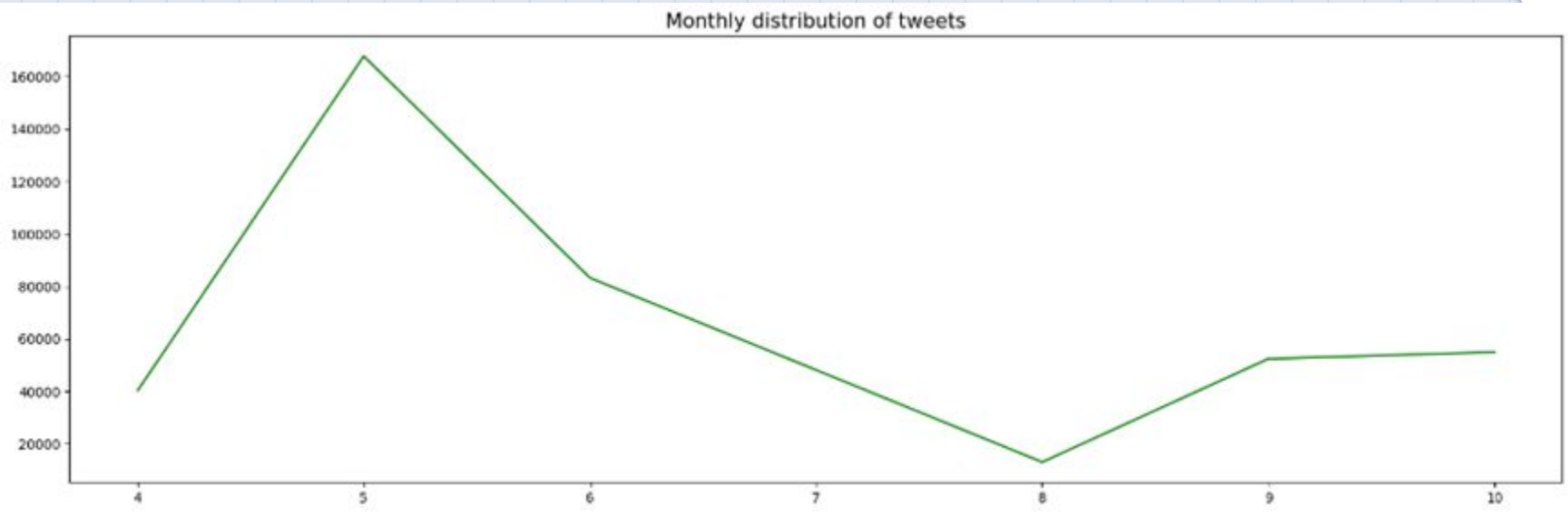
created_at	favorite_count	retweet_count	place	hashtags	user_mentions
2020-04-19	0.0	31.0	Jakarta Capital Region	NULL	GlblCtzn, priyankachopra
2020-04-19	0.0	61.0	Nigeria	NULL	OGSG_Official
2020-04-19	0.0	1.0	NULL	NULL	AdvoBarryRoux
2020-04-19	NULL	NULL	NULL	NULL	NULL
en	NULL	MobilePunch	0.0	covid19 oyo discharg two patient	0.0
2020-04-19	0.0	13869.0	NULL	Covid_19	NULL
2020-04-19	0.0	526.0	British Columbia, Canada	NULL	DrJMZimmerman
2020-04-19	0.0	119.0	London, England	NULL	NULL
2020-04-19	0.0	474.0	JPO Aesthetics	coronavirus	morethanmySLE
2020-04-19	0.0	23.0	Ottawa, Ontario	NULL	NULL
2020-04-19	0.0	6.0	NULL	NULL	TahirsyeedK, sagarikaghose
2020-04-19	0.0	92.0	NULL	NULL	JoeNBC
2020-04-19	NULL	NULL	NULL	NULL	NULL
en	NULL	BuckSexton	0.0	doctor friend deliv babi crisi told interest data larg nyc area hospit	0.7351
2020-04-19	0.0	20173.0	NULL	NULL	KatiePhang
2020-04-19	0.0	0.0	North Coast	COVID19	ClevelandClinic
2020-04-19	0.0	40.0	Kano, Nigeria.	NULL	SaharaReporters
2020-04-19	0.0	408.0			JoeConchaTV
2020-04-19	0.0	227.0			OIC_IPHRC
2020-04-19	NULL	NULL			NULL

only showing top 20 rows

clean_tweet_processed	sentiment
call leader help protect refuge covid provid qualiti health care	pos
ogun state support cbn nirsal covid target credit facil tcf	pos
polic offici base namahadi polic station busi drink liquor certain tavern whilst duti	pos
	NULL
	NULL
condol famili surviv	neu
receiv text year old son work covid patient equat quarantin com	neu
taiwan vice presid chen chien jen countri fight covid	neg
break new york woman symptom die last week prescrib drug cocktail known	neg
horribl tragedi nova scotia today famili get closur pandem funer	neu
covid cure sooner later cure ghose viru	neu
januari nd total control one person come china control go	neu
	NULL
	NULL
covid attack almost anyth bodi devast consequ feroc breathtak humbl	neg
prevent key know simpl step take today protect	pos
break jigawa record first covid case sahara report	neu
wallac pelosi also downplay covid presid underplay threat earli day speaker pel	neg
iphrc condemn unrel viciou campaign malign muslim spread	neg
	NULL

EDA on “Created_at” column

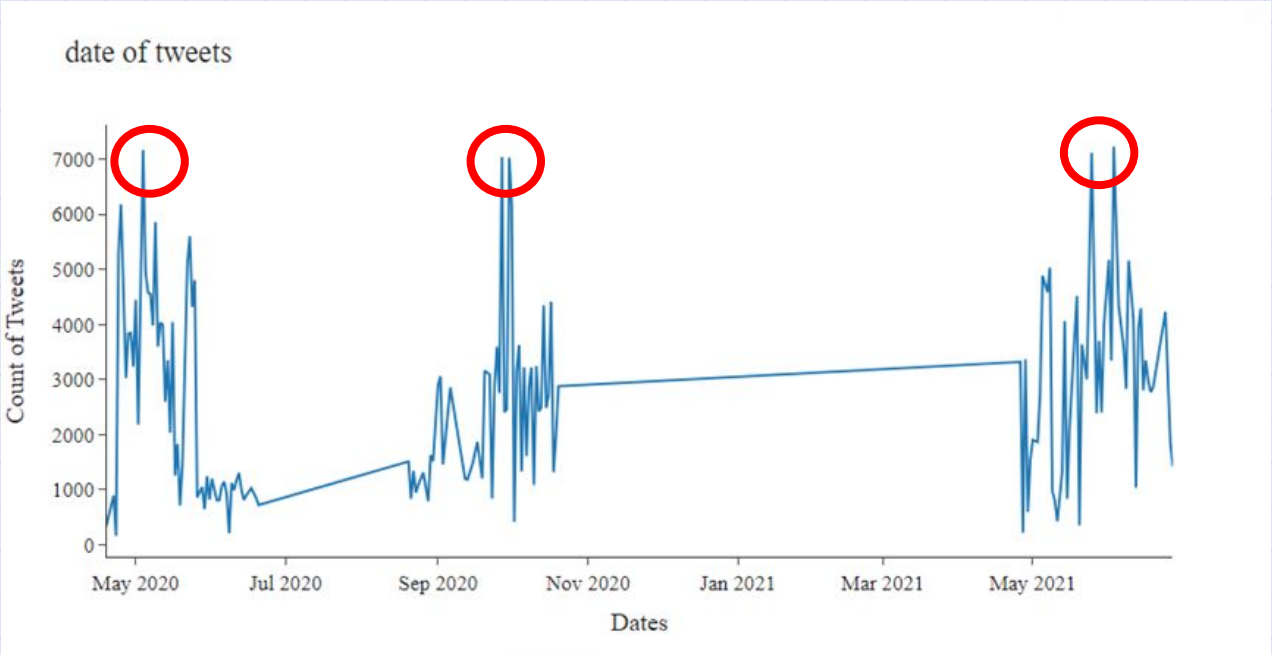
- **Created_at** stands for the time tweeted at.
- monthly distribution of tweets



➢ We can notice that date of the dataset is mostly distributed in May

EDA on “Created_at” column

- **Created_at** stands for the time tweeted at.
- Daily distribution of tweets

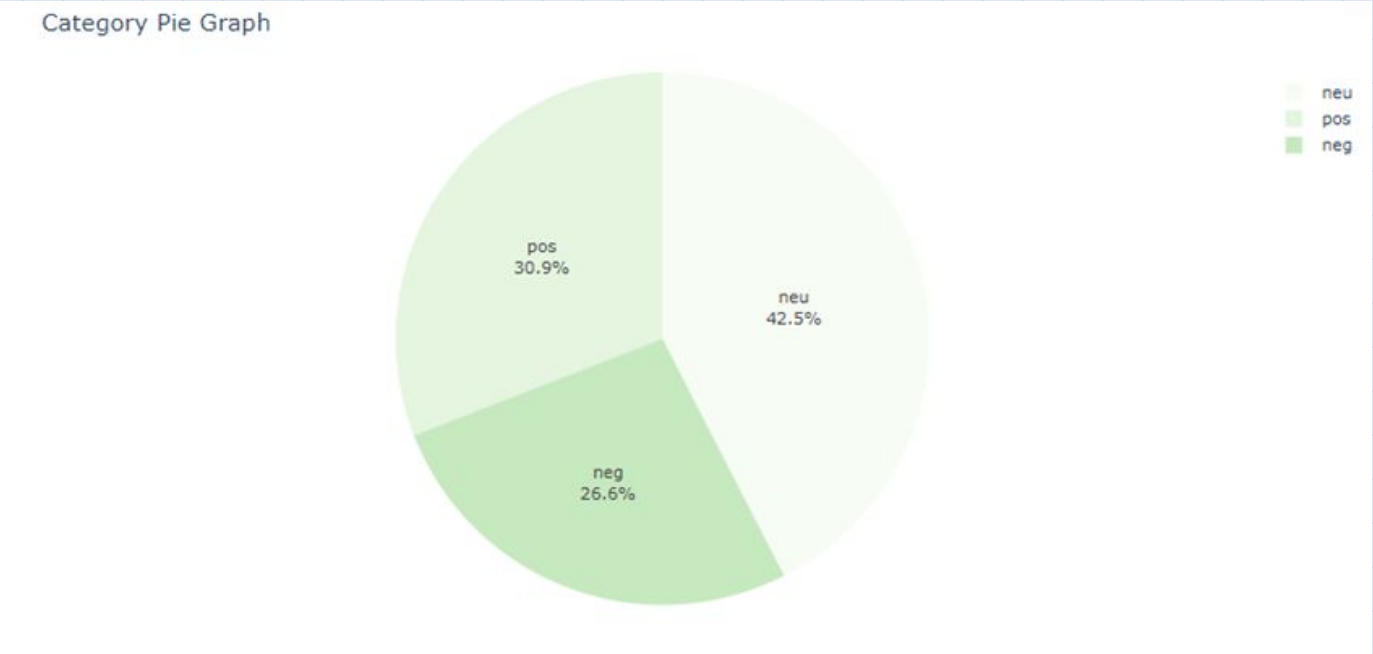


EDA on “Place” column

- **Place** stands for place of tweets
- Top 10 place of tweets:

EDA on “sentiment” column

- **Sentiments**



EDA on “favorite_count” column

created_at	favorite_count	retweet_count	clean_tweet_processed
2021-06-08	2923.0	769.0	pandem acceler innov across industri maintain pace en
2020-05-09	2020	2 new COVID-19 cases confirmed from 1	djnicejose
2020-04-30	1166.0	441.0	detroit say money month covid studi find
2021-06-05	676.0	144.0	slowli get back normal citi citi event th juli celebr
2021-05-13	571.0	100.0	support global citizen googl releas campaign spark awar encourag particip around educ
2020-08-21	554.0	94.0	latest covid protocol procedur tiu athlet date avail view websit
2021-05-04	531.0	76.0	covid new case bring total
2020-05-08	438.0	98.0	updat lot chang last six week owner focus navig busi
2020-05-12	400.0	148.0	pla partner seek new round feedback work librari respond covid crisi
2020-05-24	393.0	109.0	feel overwhelm due covid take life especi feel anxiou remem
2020-09-25	387.0	135.0	someon know math covid death alreadi count sep
2020-10-13	373.0	31.0	feed street get net talkin gt whole play stay activ covid shit go wast
2020-05-16	363.0	74.0	differ wear mask protect peopl self protect none know
2020-10-11	344.0	51.0	mr mccarthi almost new case covid day america dead america mani ame
2020-09-24	334	Health CAS Mwangangi says. https://t.co/PmfZtUUdAq"" #fb"	fb
2020-05-16	325.0	40.0	imagin bank sent sm deduct account donat fight covi
2020-05-14	313.0	30.0	covid creat disrupt safe abort medic contracept africa affect product delayi
2020-04-23	295.0	124.0	fort dix april prison four staff member confirm covid civil right
2020-09-25	270.0	104.0	covid survey halt minnesota amid racism intimid
2020-10-01	262.0	88.0	today day telehealth coverag chang wrote go effect patient ht
2020-06-17	259.0	432.0	food agricultur industri taken massiv hit due covid look innov solut across
2020-05-12	256.0	146.0	lie say get high mark handl covid real presid
2021-05-09	234.0	35.0	one forgot quebec citi gym friendli anti masker andvoc oppon restrict fine three time major
2020-10-01	211.0	40.0	amid tragic news mass graf myle mccormack version christi moor song record last month
2021-06-15	195.0	38.0	get covid vaccin render life insur polici null void vaccin regard
2020-04-25	193.0	57.0	con question feder liber covid respons continu even plumb depth cla
2021-04-30	188.0	83.0	covid warn indian health care system cater need disadvantag peopl
2020-04-29	184.0	87.0	guardian hardli sourc anyon seek evid would approach think need ex
2021-06-03	181.0	31.0	student provid evid covid vaccin statu univers confidenti
2021-05-25	178.0	25.0	white hous say unit state tuesday reach american adult fulli vaccin covid

Observing the corresponding created_at,retweet_count, and clean tweets through the "favorite_count" column.

EDA on “retweet_count” column

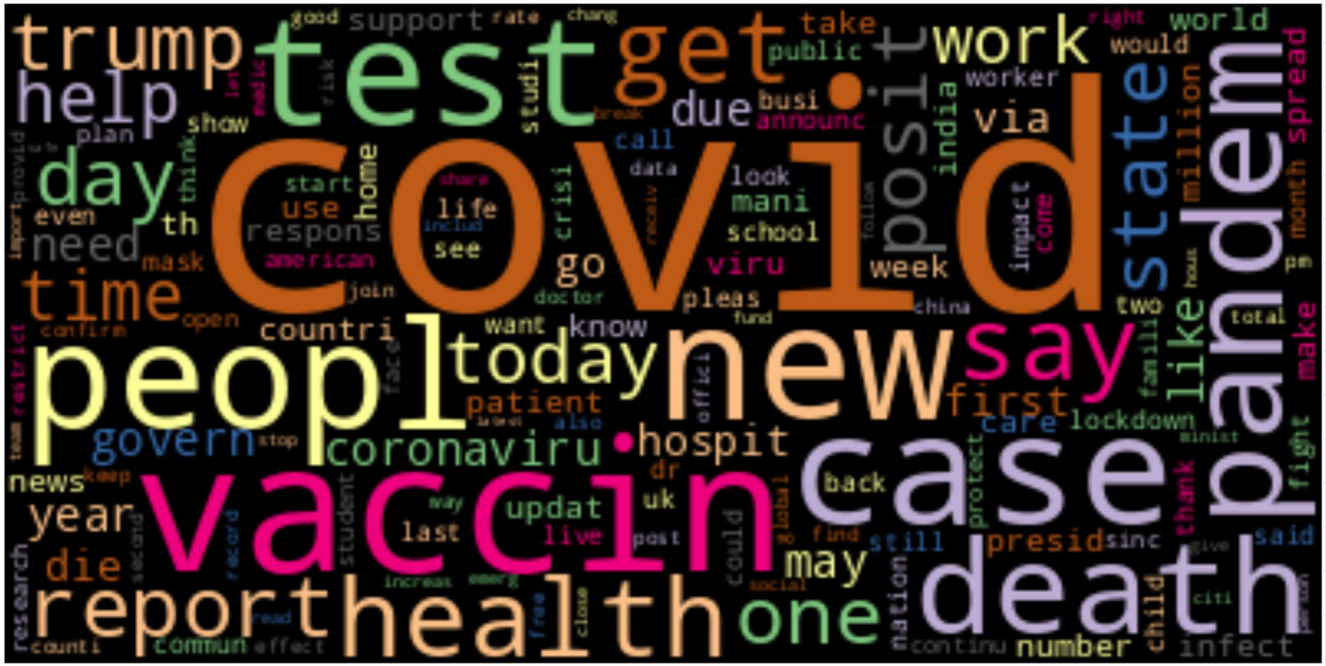
created_at	favorite_count	retweet_coun	clean_tweet_processed
2020-10-04	0.0	416923.0	addit risk death think high believ fact given potu fa
2020-10-06	0.0	416896.0	omg everi member trump debat prep team known infect except rudi giuliani
2020-10-04	0.0	416881.0	legendari japanes fashion design kenzo takada dy covid pari age
2020-10-04	0.0	416875.0	sen chuck grassley judiciari committe meet sen mike lee thursday though lee sinc test posit
2020-10-04	0.0	416474.0	school mosqu close tehran covid infect rise via
2020-10-03	0.0	414746.0	presid trump knew covid hour ago full blown coverup territori
2020-10-03	0.0	414595.0	game covid becam perfect match gamer
2020-10-03	0.0	414370.0	kellyann conway test posit covid enter quarantin
2020-10-03	0.0	414214.0	feverish fatigu presid donald trump spend weekend militari hospit treatment cov
2020-10-03	0.0	414185.0	kellyann conway test posit covid via
2020-10-03	0.0	413465.0	increas phenomenon writer groundlessli rather vicious attack anoth rudimentari covid repo
2020-10-03	0.0	412067.0	live host hear doctor eect
2020-10-03	0.0	412067.0	sewerag work supervisor use throw done ktr covid support ot one time set
2020-10-03	0.0	411960.0	biolog icmr develop covid treatment inject inactiv sar cov hors
2020-10-03	0.0	411944.0	good know driver colour heart
2020-10-03	0.0	411893.0	india pass grim mileston covid death
2020-10-03	0.0	411892.0	outrag comment come togeth nation pray presid enough disgustingli fals ac
2020-10-03	0.0	411750.0	real stori develop nation done remark better fight covid rich white
2020-10-03	0.0	411502.0	donald trump admit hospit diagnos coronaviru treatment receiv
2020-10-03	0.0	411379.0	peopl die covid age care facil aust health dept conced death cou
2020-10-03	0.0	411377.0	independ medic verifi inde covid mani sign could
2020-10-03	0.0	411369.0	covid problem case count
2020-10-03	0.0	411170.0	avoid sex kiss covid patient avoid oral sex oral kiss covid patient
2020-10-03	0.0	411170.0	strict rigid covid protocol reopen school say educ minist
2020-10-03	0.0	411169.0	whew thrill aka mom test neg pray still await result strug
2020-10-03	0.0	411168.0	recent exampl show real threat cavali attitud toward ver
2020-04-29	0.0	399220.0	top ny er doctor treat covid patient dy suicid
2020-04-30	0.0	399097.0	landlord tenant struggl rental mortgag payment due told nifti map
2020-05-01	0.0	399005.0	break tennesse dept health report new covid case addit death sinc thursday
2020-10-02	0.0	389339.0	break notr dame presid fr john jenkin attend ami coney barrett nomin white hous satu

Observing the corresponding created_at, favorite_count, and clean tweets through the "retweet_count" column.

EDA on “clean_tweet_processed”

word	count
covid	167941
vaccin	26181
new	21296
case	20373
test	18025
peopl	17675
pandem	13527
death	13048
health	12641
get	11146
say	10823
report	10720
one	9948
trump	9532
day	9462
posit	9353
state	8998
today	8898
time	8485
help	8299

only showing top 20 rows



Analysis of the "clean_tweet_processed" for all the data.

The evolving tech world

Yearly

word count of 2020

word	count_2020
covid	109628
test	14226
new	13894
case	13239
peopl	11654
death	9143
trump	8735
pandem	8681
health	7871
say	7303
posit	7212
report	6697
one	6599
day	6245
coronaviru	6155
get	6021
state	5963
time	5763
today	5636
help	5594

only showing top 20 rows

word count of 2021

word	count_2021
covid	58046
vaccin	21873
new	7325
case	7019
peopl	5992
get	5108
pandem	4831
health	4752
india	4027
report	3985
death	3846
test	3753
say	3515
one	3338
today	3216
day	3188
may	3176
year	3061
state	3016
first	2816

only showing top 20 rows

Monthly

Top 5 words in 2020 each months:

month	word	count
4	covid	13010
4	new	1486
4	test	1427
4	peopl	1418
4	death	1209
5	covid	41220
5	new	4775
5	case	4379
5	test	4370
5	peopl	4260
6	covid	7116
6	test	902
6	case	900
6	new	887
6	peopl	779
8	covid	5049
8	case	763
8	test	703
8	new	684
8	peopl	561
9	covid	20585
9	test	2999
9	case	2963
9	new	2857
9	peopl	2298

only showing top 25 rows

Top 5 words in 2021 each months:

month	word	count
4	covid	3244
4	vaccin	1114
4	india	641
4	case	394
4	peopl	357
5	covid	27102
5	vaccin	9890
5	new	3272
5	case	3252
5	peopl	2845
6	covid	27700
6	vaccin	10869
6	new	3712
6	case	3373
6	peopl	2790

Analysis of the "clean_tweet_processed" according to the year and month.

EDA on “hashtags” column

Most Frequent Hashtags of data

hashtag	count
COVID19	8632
coronavirus	4683
COVID	4173
Covid_19	3293
Covid	2457
covid	1429
lockdown	1251
Coronavirus	1250
India	1032
COVID__19	966
Covid19	953
covid19	831
StaySafe	754
0.0	657
BREAKING	514
KeepingOurPromiseAct	464
HR3548	461
USA	440
pandemic	401
StayHome	378

only showing top 20 rows

Analysis of the "hashtags" for all the data.

EDA on “hashtags” column

Yearly

hashtags count of 2020

hashtag	count_2020
COVID19	5698
coronavirus	3415
Covid_19	2870
COVID	2222
Covid	997
COVID__19	959
Coronavirus	885
lockdown	644
Covid19	637
covid	614
covid19	529
0.0	420
StaySafe	417
BREAKING	289
COVID—19	278
COVID_19	273
pandemic	249
Trump	218
SocialDistancing	188
staysafe	183

only showing top 20 rows

hashtags count of 2021

hashtag	count_2021
COVID19	2906
COVID	1937
Covid	1455
coronavirus	1263
covid	813
lockdown	606
KeepingOurPromiseAct	464
HR3548	461
Covid_19	415
Coronavirus	359
StaySafe	335
Covid19	315
India	282
COVISHIELD	236
vaccine	234
BREAKING	223
StayHome	219
0.0	201
covid19	189
China	172

only showing top 20 rows

Monthly

Top 5 words in 2020 each months:

month	hashtag	count
4	Covid_19	718
4	COVID19	699
4	coronavirus	235
4	COVID	178
4	COVID__19	75
5	COVID19	2042
5	Covid_19	1446
5	COVID	560
5	coronavirus	547
5	COVID__19	467
6	COVID19	387
6	Covid_19	127
6	COVID	83
6	COVID__19	75
6	coronavirus	68
8	coronavirus	272
8	COVID19	248
8	COVID	156
8	Coronavirus	82
8	Covid_19	73
9	COVID19	1211
9	coronavirus	1160
9	COVID	550
9	Covid	361
9	Coronavirus	286

only showing top 25 rows

Top 5 words in 2021 each months:

month	hashtag	count
4	COVID19	176
4	Covid	132
4	COVID	121
4	coronavirus	76
4	India	52
5	COVID19	1393
5	COVID	968
5	Covid	835
5	coronavirus	628
5	covid	441
6	COVID19	1337
6	COVID	848
6	coronavirus	559
6	Covid	488
6	KeepingOurPromiseAct	464

Analysis of the "hashtags" according to the year and month.

EDA on “mention” column

Most Frequent mention of data

mention	count
realDonaldTrump	2956
YouTube	1032
Reuters	937
narendramodi	801
SkyNews	765
CNN	716
WHO	607
BorisJohnson	597
JoeBiden	569
ABC	546
POTUS	523
thehill	509
PMOIndia	404
PTI_News	395
WhiteHouse	392
nytimes	381
RepRitchie	378
CDCgov	374
TheEconomist	371
NBCNews	370

only showing top 20 rows

Analysis of the "mention" for all the data.

EDA on “mention” column

Yearly

mentions count of 2020

mention	count_2020
realDonaldTrump	2943
YouTube	706
CNN	544
JoeBiden	482
Reuters	468
SkyNews	464
narendramodi	422
BorisJohnson	402
ABC	387
WHO	372
thehill	339
guardian	313
NBCNews	278
TheEconomist	268
MattHancock	256
nowthisnews	252
CBSNews	252
business	245
NYGovCuomo	242
nytimes	234

only showing top 20 rows

mentions count of 2021

mention	count_2021
Reuters	469
narendramodi	378
RepRitchie	378
YouTube	326
POTUS	310
SkyNews	301
WHO	235
BorisJohnson	194
WhiteHouse	188
PTI_News	187
thewire_in	182
PMOIndia	181
HouseDemocrats	177
CDCgov	172
DASimmigration	172
CNN	171
thehill	170
IndiaToday	164
ABC	159
TravelGov	150

only showing top 20 rows

Monthly

Top 5 mentions in 2020 each months:

month	mention	count
4	realDonaldTrump	332
4	YouTube	135
4	ABC	71
4	CNN	70
4	WHO	67
5	realDonaldTrump	867
5	YouTube	308
5	CNN	171
5	narendramodi	162
5	Reuters	144
6	realDonaldTrump	135
6	CNN	41
6	narendramodi	40
6	WHO	27
6	guardian	26
8	realDonaldTrump	96
8	narendramodi	40
8	JoeBiden	38
8	PMOIndia	33
8	Reuters	32
9	realDonaldTrump	467
9	JoeBiden	155
9	BorisJohnson	135
9	SkyNews	122
9	Reuters	98

only showing top 25 rows

Top 5 mentions in 2021 each months:

month	mention	count
4	narendramodi	49
4	Reuters	32
4	PTI_News	23
4	WHO	23
4	timesofindia	19
5	Reuters	242
5	narendramodi	218
5	YouTube	159
5	thewire_in	141
5	SkyNews	131
6	RepRitchie	378
6	Reuters	195
6	HouseDemocrats	174
6	POTUS	166
6	WhiteHouse	155

Analysis of the "mention" according to the year and month.

04

Text Clustering using k-means and PCA

Text Clustering — k-means

- **k-means clustering** is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.

- source: wikipedia
  https://en.wikipedia.org/wiki/K-means_clustering

Code and outcome

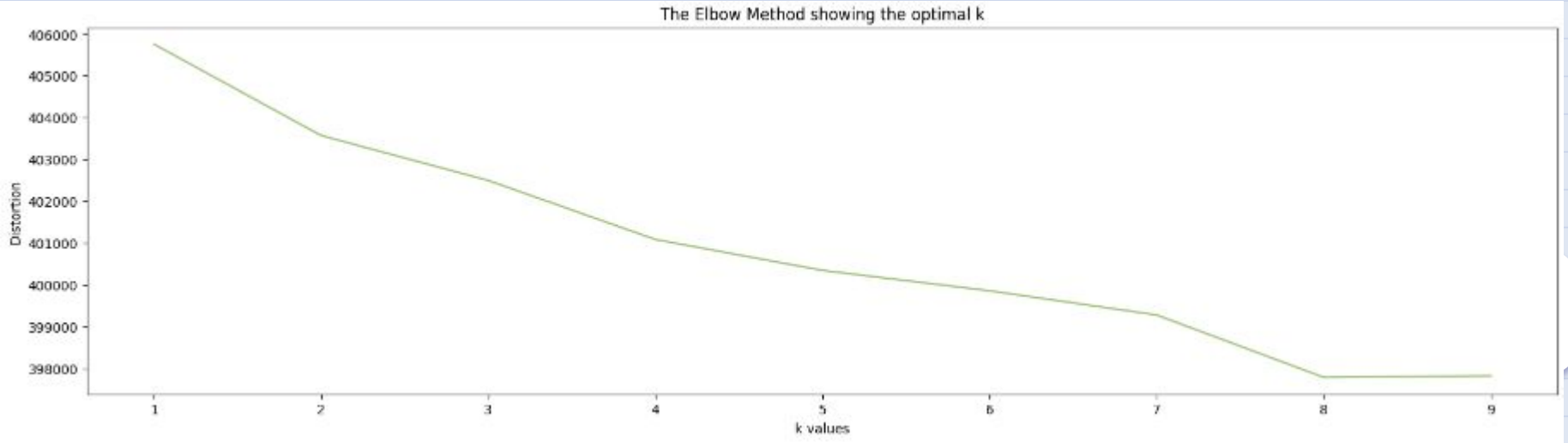
Text Clustering — k-means

Elbow method is to select the optimal number of clusters by fitting the model with a range of values for k.

```
distortions = []
K = range(1,10)
for k in K:
    kmean = KMeans(n_clusters=k,random_state=7)
    kmean.fit(X)
    distortions.append(kmean.inertia_)

plt.figure(figsize=(20,5))
plt.plot(K, distortions, '-',color='g')
plt.xlabel('k values')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```

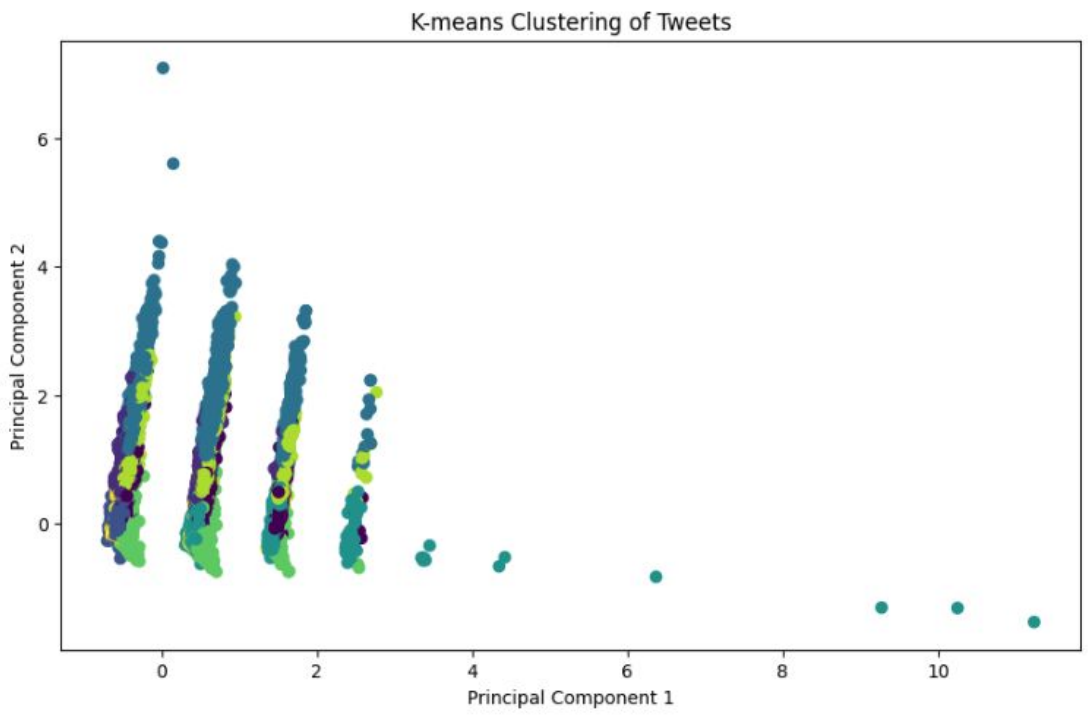
Text Clustering — k-means

Elbow method is to select the optimal number of clusters by fitting the model with a range of values for k.



➢ **Best: k = 8**

Text Clustering — k-means



➢ **Best: k = 8**

Text Clustering — PCA

- **Principal component analysis (PCA)** is the process of computing the principal components and using them to perform a change of basis on the data,sometimes using only the first few principal components and ignoring the rest.

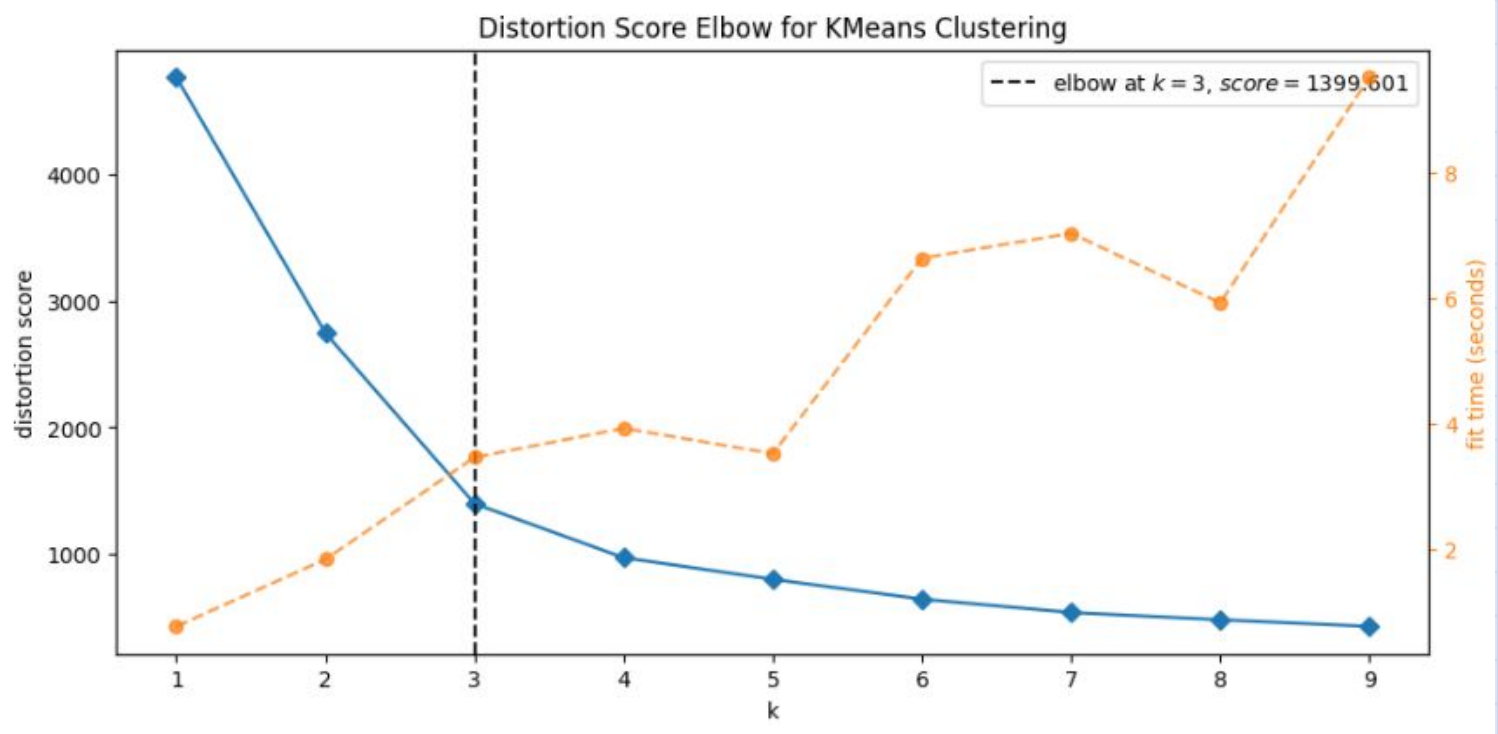
- source: wikipedia
https://en.wikipedia.org/wiki/Principal_component_analysis

Text Clustering — k-means After PCA

```python
from sklearn.decomposition import IncrementalPCA
from tqdm import tqdm


# 創建 IncrementalPCA 實例
inc_pca = IncrementalPCA(n_components=2)

# 逐批次進行 PCA 並顯示進度條
batch_size = 1000
num_batches = X.shape[0] // batch_size + 1

# 初始化結果矩陣
X_pca = np.zeros((X.shape[0], 2))

for i in tqdm(range(num_batches), desc="PCA Progress"):
    start_idx = i * batch_size
    end_idx = min((i + 1) * batch_size, X.shape[0])
    X_batch = X[start_idx:end_idx, :].toarray()

    # 在每個批次上進行擬合
    inc_pca.partial_fit(X_batch)

    # 同時進行 transform
    X_pca[start_idx:end_idx, :] = inc_pca.transform(X_batch)
```

```
PCA Progress: 100%|██████████| 412/412 [29:02<00:00,  4.23s/it]
```

Text Clustering — k-means After PCA



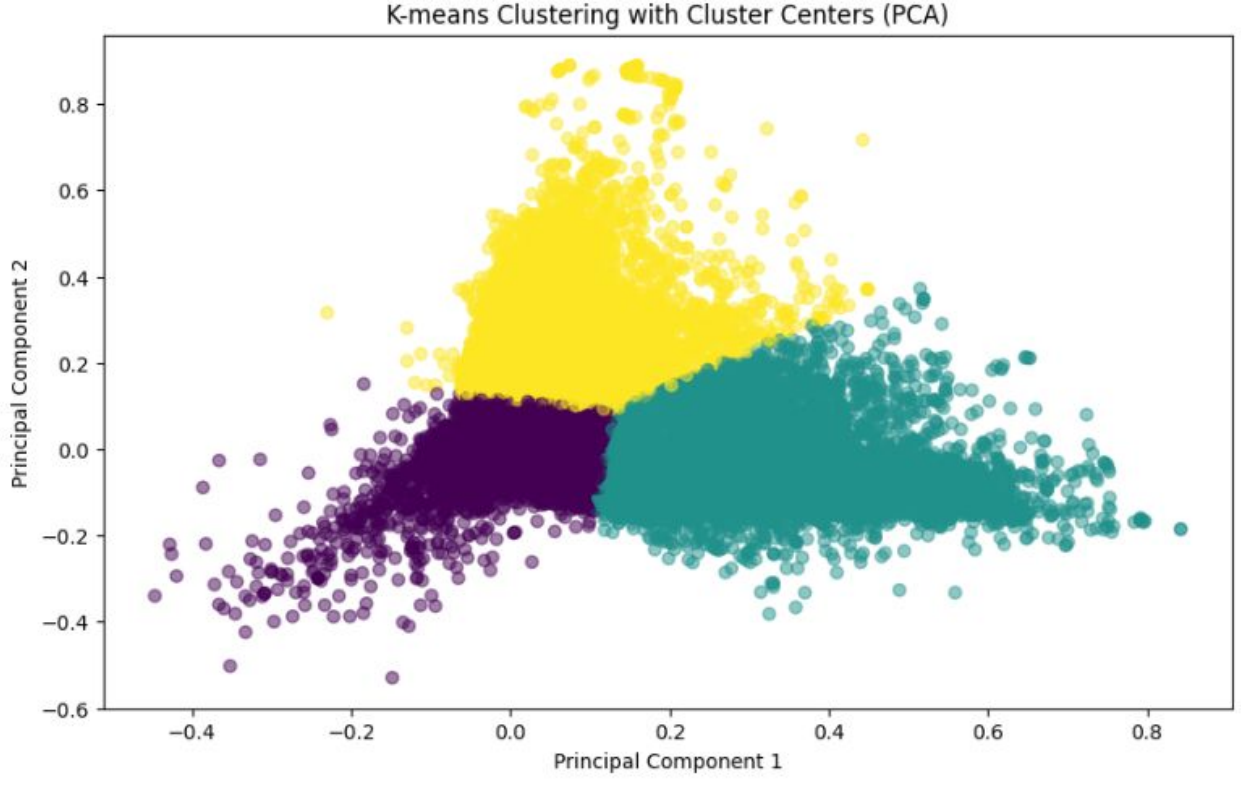
➢ Best: k = 3

Text Clustering — k-means After PCA

```
# 重新適配 K-means 模型
visualizer_pca = KElbowVisualizer(KMeans(), k=(1, 10), size=(1080, 500))
visualizer_pca.fit(X_pca)  # 使用 PCA 後的資料進行擬合
optimal_k_pca = visualizer_pca.elbow_value_  # 取得 PCA 後的最佳聚類數目

# 使用 PCA 後的最佳聚類數目重新適配 K-means 模型
kmeans_pca = KMeans(n_clusters=optimal_k_pca, random_state=42)
df['cluster_pca'] = kmeans_pca.fit_predict(X_pca)
```

- Best: k = 3

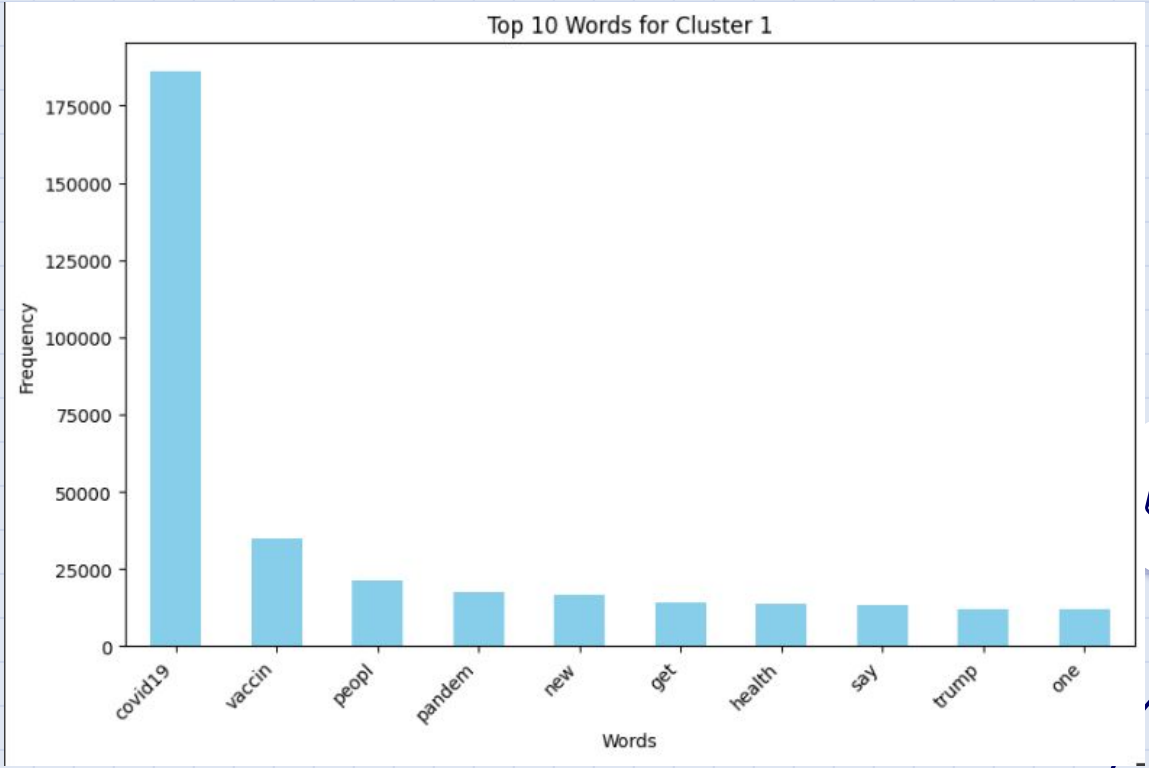
Text Clustering — k-means After PCA



Text Clustering — k-means After PCA

Top words for Cluster 1

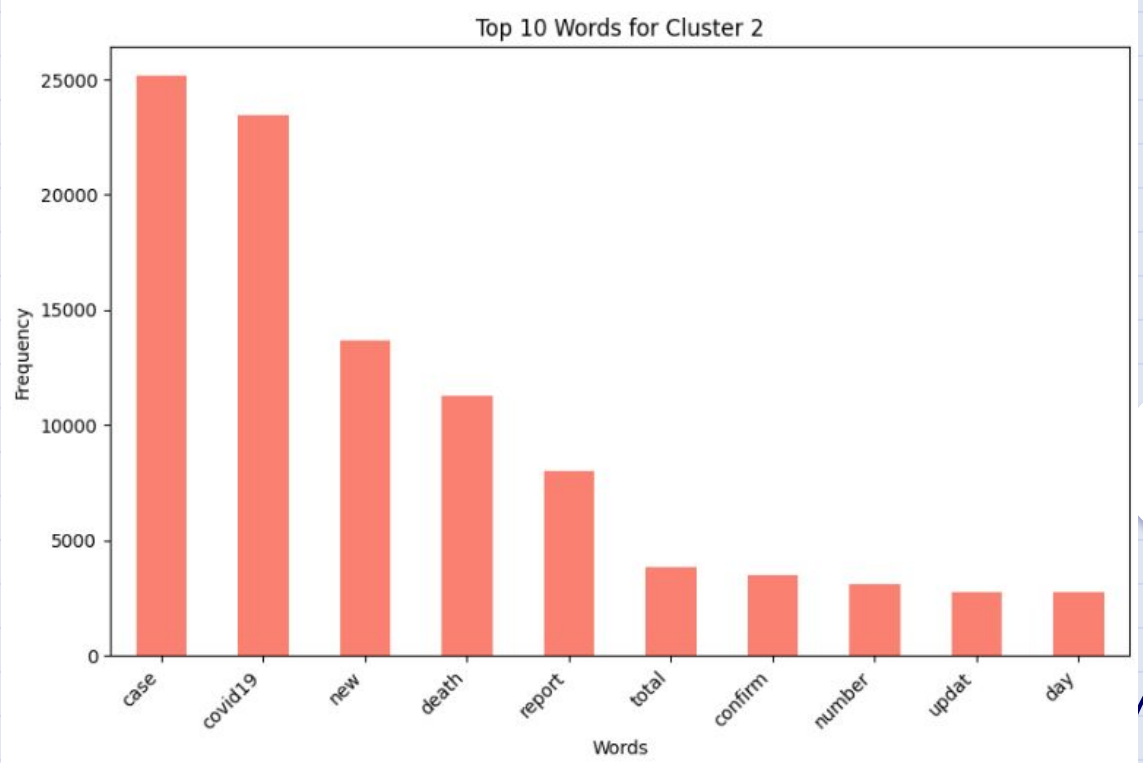
covid19	185983
vaccin	34669
peopl	21222
pandem	17406
new	16689
get	13980
health	13503
say	13176
trump	12056
one	11904



Text Clustering — k-means After PCA

Top words for Cluster 2

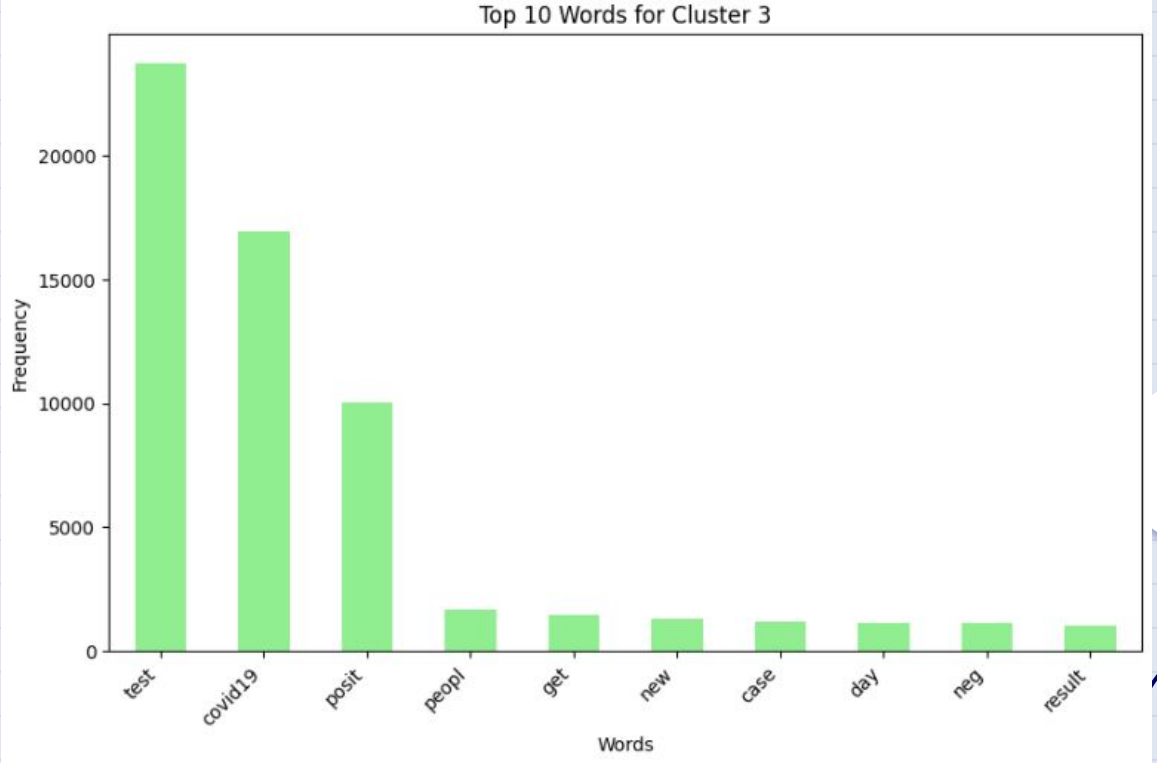
case	25155
covid19	23422
new	13687
death	11246
report	8018
total	3839
confirm	3496
number	3087
updat	2741
day	2735



Text Clustering — k-means After PCA

Top words for Cluster 2

test	23732
covid19	16956
posit	10018
peopl	1651
get	1440
new	1305
case	1211
day	1140
neg	1118
result	1029



Text Clustering

- **Topic of Cluster1**
  It may be a **comprehensive discussion cluster**, focusing on various aspects of public concern about COVID-19. In addition to vaccines and the pandemic, it also includes topics related to health, speech, and figures like Trump.

- **Topic of Cluster2**
  This cluster may be more focused on **pandemic data and statistics**. In addition to cases, new discoveries, and deaths, it includes terms related to reporting, totals, confirmations, numbers, and updates, indicating discussions about pandemic data trends.

- **Topic of Cluster3**
  It could be a specialized discussion cluster on **COVID-19 testing**, covering test results, people's reactions to testing, infection cases, and the nature of results (positive and negative).

05

Sentiment Prediction

Reference

A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis

Published: 05 May 2021

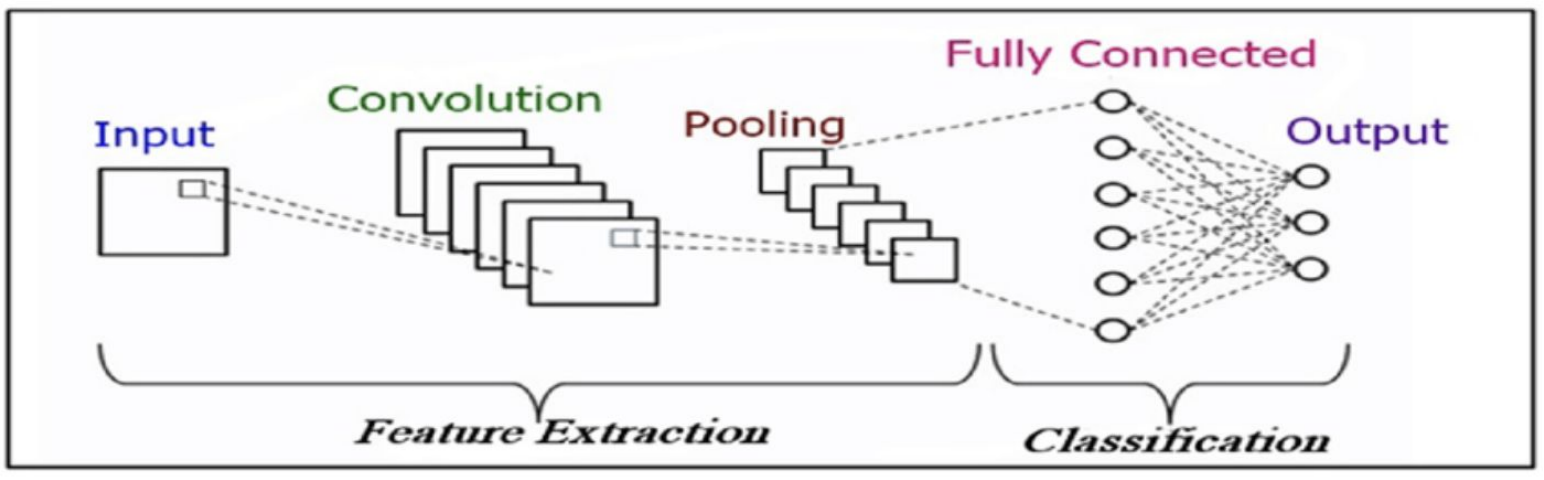
Volume 77, pages 13911–13932, (2021) Cite this article

Abstract

As the number of users getting acquainted with the Internet is escalating rapidly, there is more user-generated content on the web. Comprehending hidden opinions, sentiments, and emotions in emails, tweets, reviews, and comments is a challenge and equally crucial for social media monitoring, brand monitoring, customer services, and market research. Sentiment analysis determines the emotional tone behind a series of words may essentially be used to understand the attitude, opinions, and emotions of users. We propose a novel long short-term memory (LSTM)–convolutional neural networks (CNN)–grid search-based deep neural network model for sentiment analysis. The study considers baseline algorithms like convolutional neural networks, *K*-nearest neighbor, LSTM, neural networks, LSTM–CNN, and CNN–LSTM which have been evaluated using accuracy, precision, sensitivity, specificity, and F-1 score, on multiple datasets. Our results show that the proposed model based on hyperparameter optimization outperforms other baseline models with an overall accuracy greater than 96%.

When Model Building

CNN



Reference:
https://www.researchgate.net/figure/The-block-diagram-of-CNN-architecture-A-CNN-model-automatically-learns-spatial-feature_fig3_364730045 [accessed 27 Dec, 2023]

CNN

```python
import torch.nn.functional as F

# simple CNN model
class SimpleCNN(nn.Module):
    def __init__(self):
        super(SimpleCNN, self).__init__()
        self.conv1 = nn.Conv1d(in_channels=1, out_channels=64, kernel_size=3)
        self.pool = nn.MaxPool1d(kernel_size=2)
        self.fc1 = nn.Linear(64 * 499, 3)  # 3 is the number of categories

    def forward(self, x):
        # Reshape input
        x = x.view(x.size(0), 1, -1)
        x = self.pool(F.relu(self.conv1(x)))
        x = x.view(x.size(0), -1)
        x = self.fc1(x)
        return x


# initialize the model, loss function, and optimizer
model = SimpleCNN()
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

# model training
epochs = 10
for epoch in range(epochs):
    for inputs, labels in train_dataloader:
        optimizer.zero_grad()
        # Add one dimension to the second dimension, in order to meet the requirements of the convolutional layer
        outputs = model(inputs.unsqueeze(1))
        # turn the labels into long type
        loss = criterion(outputs, labels.squeeze().long())
        loss.backward()
        optimizer.step()
```

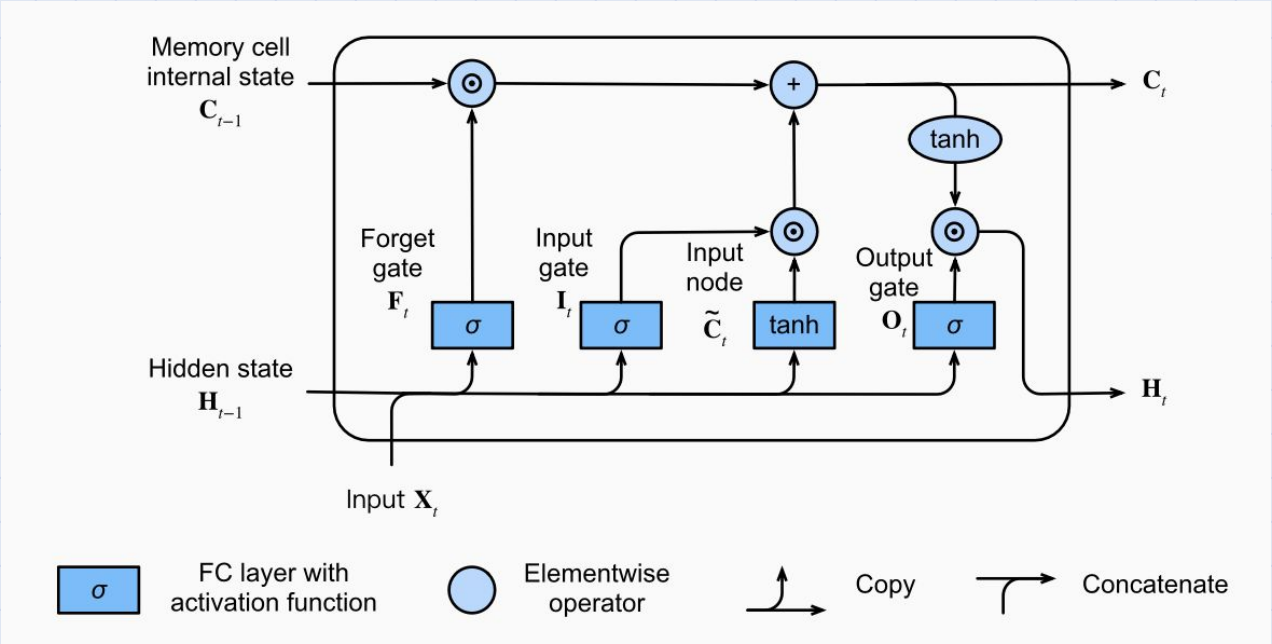
```python
from sklearn.metrics import accuracy_score

# 计算准确度
accuracy = accuracy_score(true_labels, predicted_labels)

print(f"Test Accuracy: {accuracy * 100:.2f}%")
```

```
Test Accuracy: 86.12%
```

The CNN accuracy in this test dataset is 86.12%

When Model Building

LSTM



Reference: Architecture of a LSTM Unit (Credits: https://d2l.ai/chapter_recurrent-modern/lstm.html)

LSTM

```python
class SimpleLSTM(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(SimpleLSTM, self).__init__()
        self.lstm = nn.LSTM(input_size, hidden_size, batch_first=True)
        self.fc = nn.Linear(hidden_size, output_size)

    def forward(self, x):
        # need to let the shape of x be (batch_size * sequence_length, input_size)
        x = x.view(x.size(0), -1, x.size(-1))

        _, (h_n, _) = self.lstm(x)
        out = self.fc(h_n[-1, :, :])
        return out


# initialize the model, loss function, and optimizer
input_size = 1000
hidden_size = 64
output_size = 3
model = SimpleLSTM(input_size, hidden_size, output_size)
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)


# model training
epochs = 10
for epoch in range(epochs):
    model.train()

    running_loss = 0.0
    for inputs, labels in train_dataloader:
        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(outputs, labels.squeeze().long())
        loss.backward()
        optimizer.step()

        running_loss += loss.item()

    # print the training loss
    print(f"Epoch {epoch + 1}/{epochs}, Training Loss: {running_loss / len(train_dataloader)}")
```

```
Epoch 1/10, Training Loss: 0.47494390922043894
Epoch 2/10, Training Loss: 0.4158572810115784
Epoch 3/10, Training Loss: 0.4008712821358431
Epoch 4/10, Training Loss: 0.3840216678253005
Epoch 5/10, Training Loss: 0.36546466442892855
Epoch 6/10, Training Loss: 0.3434408057669635
Epoch 7/10, Training Loss: 0.31960032704359587
Epoch 8/10, Training Loss: 0.2950940129356772
Epoch 9/10, Training Loss: 0.27177009155458887
Epoch 10/10, Training Loss: 0.2500467841174545
```

```python
# 计算准确度
lstm_accuracy = accuracy_score(lstm_true_labels, lstm_predicted_labels)

print(f"LSTM Test Accuracy: {lstm_accuracy * 100:.2f}%")
```

```
LSTM Test Accuracy: 85.42%
```

The LSTM accuracy in this test dataset is 85.42%

CNN-LSTM

```python
import torch.nn as nn
import torch.optim as optim
import torch.nn.functional as F

class CombinedModel(nn.Module):
    def __init__(self, cnn_input_size, lstm_input_size, hidden_size, output_size):
        super(CombinedModel, self).__init__()

        # CNN layer
        self.conv1 = nn.Conv1d(in_channels=1, out_channels=64, kernel_size=3)
        self.pool = nn.MaxPool1d(kernel_size=2)
        self.fc_cnn = nn.Linear(64 * cnn_input_size, lstm_input_size)

        # LSTM layer
        self.lstm = nn.LSTM(lstm_input_size, hidden_size, batch_first=True)

        # Fully connected layer for sentiment prediction
        self.fc_final = nn.Linear(hidden_size, output_size)

    def forward(self, x):
        # CNN layer
        x_cnn = x.view(x.size(0), 1, -1)
        x_cnn = self.pool(F.relu(self.conv1(x_cnn)))
        x_cnn = x_cnn.view(x_cnn.size(0), -1)
        x_cnn = F.relu(self.fc_cnn(x_cnn))

        # LSTM layer
        x_lstm, _ = self.lstm(x_cnn.unsqueeze(1))

        # Fully connected layer for sentiment prediction
        x_final = self.fc_final(x_lstm[:, -1, :])

        return x_final

# Initialize the model, criterion, and optimizer
cnn_input_size = 499  # Change this based on the output size of your CNN layer
lstm_input_size = 64  # Change this based on the hidden size of your LSTM layer
hidden_size = 64      # Change this based on your requirements
output_size = 3       # Number of sentiment classes

combined_model = CombinedModel(cnn_input_size, lstm_input_size, hidden_size, output_
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(combined_model.parameters(), lr=0.001)

# Train the combined model
epochs = 10
for epoch in range(epochs):
    # Set the model to training mode
    combined_model.train()

    running_loss = 0.0
    for inputs, labels in train_dataloader:
        optimizer.zero_grad()
        outputs = combined_model(inputs)
        loss = criterion(outputs, labels.squeeze().long())
        loss.backward()
        optimizer.step()

        running_loss += loss.item()

    # Print the training loss
    print(f"Epoch {epoch + 1}/{epochs}, Training Loss: {running_loss / len(train_dataloader)}")
```

```
Epoch 1/10, Training Loss: 0.5252266864529536
Epoch 2/10, Training Loss: 0.4093809833687432
Epoch 3/10, Training Loss: 0.397923984642595
Epoch 4/10, Training Loss: 0.3890251283707399
Epoch 5/10, Training Loss: 0.3811074899755894
Epoch 6/10, Training Loss: 0.3733362336626996
Epoch 7/10, Training Loss: 0.36588812790951264
Epoch 8/10, Training Loss: 0.3599150890073219
Epoch 9/10, Training Loss: 0.3532913015212932
Epoch 10/10, Training Loss: 0.3474617716909371
```

```python
from sklearn.metrics import accuracy_score

# count accuracy
COM_accuracy = accuracy_score(com_true_labels, com_predicted_labels)

print(f"Combined Model Test Accuracy: {COM_accuracy * 100:.2f}%")
```

```
Combined Model Test Accuracy: 86.46%
```

The CNN-LSTM accuracy in this test dataset is 86.46%

CNN

```
from sklearn.metrics import accuracy_score

# 计算准确度
accuracy = accuracy_score(true_labels, predicted_labels)

print(f"Test Accuracy: {accuracy * 100:.2f}%")
```

```
Test Accuracy: 86.12%
```

LSTM

```
# 计算准确度
lstm_accuracy = accuracy_score(lstm_true_labels, lstm_predicted_labels)

print(f"LSTM Test Accuracy: {lstm_accuracy * 100:.2f}%")
```

```
LSTM Test Accuracy: 85.42%
```

CNN-LSTM

```
from sklearn.metrics import accuracy_score

# count accuracy
COM_accuracy = accuracy_score(com_true_labels, com_predicted_labels)

print(f"Combined Model Test Accuracy: {COM_accuracy * 100:.2f}%")
```

```
Combined Model Test Accuracy: 86.46%
```

In this dataset, CNN-LSTM model would be a little better.

06

Conclusion

Analysis of Tweet Topics:

Observation of Date Distribution:

- Speculation that spring and autumn are peak periods, possibly indicating flu seasons.
- Conclusion: The pandemic may have a more noticeable impact during these seasons.

Regional Analysis:

- The United States has the highest number of tweets, suggesting a severe impact of the pandemic.
- Conclusion: The U.S. may be a hotspot for the pandemic.

Sentiment Analysis:

- Tweets predominantly exhibit neutral and positive sentiments, indicating overall stability and positivity during the pandemic.
- Conclusion: Overall sentiment tends to be positive, suggesting a proactive response to the pandemic.

Focus on Topics:

- Regardless of the topic, it seems that everyone is most concerned about issues related to COVID.
- Conclusion: COVID remains a societal focal point, with high levels of attention.

Twitter Text Analysis:

Twitter Text Frequency Analysis:

- Analyzing the most frequently used words during the pandemic to understand the main content of tweets.
- **Conclusion:** Text analysis reveals key content and discussion topics.

Mentioned Text Frequency Analysis:

- Understanding the level of attention to certain people or things through the mentioned words.
- **Conclusion:** Mentioned words provide insights into what people are focusing on and discussing.

Hashtags Text Frequency Analysis:

- Understanding how hashtags are used to increase tweet visibility.
- **Conclusion:** Hashtags reflect the expectations on social media regarding pandemic-related topics.

Future Recommendations:

Optimization of Sentiment Model:

- Optimize sentiment models for various topics to analyze emotional tendencies in different areas.
- **Objective:** Gain a comprehensive understanding of emotional changes in different topics.

Analysis of Global Health-Related Topics:

- Focus on global health-related topics for further research into societal concerns about health and public safety.
- **Objective:** Provide in-depth analysis of health-related issues for insights into future similar situations.

In summary, the analysis provides insights into societal emotions, key concerns, and discussions during the pandemic. This information aids in better understanding societal reactions and needs, offering guidance for future response measures.

Thanks!