Sentiment Analysis for Latest Korean Drama

The crawling process was done using Tweet-Harvest.

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import re
5 import requests
```

```
1 !pip install sastrawi -q
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 209.7/209.7 kB 5.4 MB/s eta 0:00:00
```

Twitter Auth Token

```
1 #@title Twitter Auth Token
2
3 twitter_auth_token = 'ac0d89e461873ad16eeb27eafca73d7d31816500'
```

```
# Import required Python package
!pip install pandas

# Install Node.js (because tweet-harvest built using Node.js)
!sudo apt-get update
!sudo apt-get install -y ca-certificates curl gnupg
!sudo mkdir -p /etc/apt/keyrings
!curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg

!NODE_MAJOR=20 && echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro

!sudo apt-get update
!sudo apt-get install nodejs -y

!node -v
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.25.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.
Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease [3,626 B]
Get:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  InRelease [1,581 B]
Get:3 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  Packages [802 kB]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:7 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy InRelease
Hit:8 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Hit:9 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy InRelease
Hit:10 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Hit:11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1,081 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,920 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,641 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,358 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [61.2 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2,107 kB]
```

```
Fetched 9,205 kB in 2s (3,745 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
curl is already the newest version (7.81.0-1ubuntu1.16).
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 45 not upgraded.
deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_20.x nodistro main
Hit:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
Hit:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  InRelease
Get:3 https://deb.nodesource.com/node_20.x nodistro InRelease [12.1 kB]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://deb.nodesource.com/node_20.x nodistro/main amd64 Packages [5,901 B]
Hit:7 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:8 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:9 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy InRelease
Hit:10 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Hit:11 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy InRelease
Hit:12 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Fetched 18.0 kB in 1s (14.4 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 45 not upgraded.
Need to get 31.6 MB of archives.
After this operation, 196 MB of additional disk space will be used.
```

Crawl Twitter Data

```
# Crawl Twitter Data

filename = 'marry_my_husband_review'
search_keyword = 'marry my husband lang:id'
limit = 2000
```

```
!npx --yes tweet-harvest -o "{filename}" -s "{search_keyword}" -l {limit} --token {twitter_auth_token}
```

```
-- Scrolling... (1) (2)

Your tweets saved to: /content/tweets-data/marry_my_husband_review.csv
Total tweets saved: 171


Your tweets saved to: /content/tweets-data/marry_my_husband_review.csv
Total tweets saved: 175


Your tweets saved to: /content/tweets-data/marry_my_husband_review.csv
Total tweets saved: 178


Your tweets saved to: /content/tweets-data/marry_my_husband_review.csv
Total tweets saved: 180


Your tweets saved to: /content/tweets-data/marry_my_husband_review.csv
Total tweets saved: 181

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21)No more tweet
Timeout reached 1 times, making sure again...

-- Scrolling... (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21)No more tweet
```

```
import pandas as pd

filename = 'marry_my_husband_review.csv'
file_path = f"/content/tweets-data/{filename}"  #

df = pd.read_csv(file_path) #csv no need delimiter
display(df)

```

	conversation_id_str	created_at	favorite_count	full_text	id_str	
0	1773327469375840530	Thu Mar 28 12:33:10 +0000 2024	48	Beautiful Kang Yoomi #ParkMinYoung #MarryMyHus...	1773327469375840530	https://pbs.twimg.com/media/GJwf7Yebc
1	1775588248032849958	Wed Apr 03 18:16:41 +0000 2024	68	Kang Jiwon's cute paparazzi #유지혁 #강지원 #나인우 #박민...	1775588248032849958	https://pbs.twimg.com/media/GKQoF92b
2	1774809727228317847	Mon Apr 01 14:55:48 +0000 2024	3014	@kdrama_menfess Yang jadi sumin di marry my hu...	1774812915516031155	
3	1773395130994807039	Thu Mar 28 17:02:01 +0000 2024	0	Comel dahla matching color baju eee #marrymyhu...	1773395130994807039	https://pbs.twimg.com/ext_tw_video_thu
4	1774626494666731679	Mon Apr 01 11:19:30 +0000 2024	113	@kdrama_menfess Aku prefer suminah (marry my h...	1774758482631377134	
...	...	...	...	...	...	
176	1775130109885940152	Tue Apr 02 11:56:13 +0000 2024	2	Terseret Skandal Kekerasan Foto SD Song Ha Yoo...	1775130109885940152	
177	1774815639833862507	Mon Apr 01 15:11:43 +0000 2024	1	@Hsnmub @tang__kira Fight for My Way jadi teme...	1774816922976260363	
178	1773959621994279020	Sat Mar 30 12:11:59 +0000 2024	0	@Soletil @Jaeyoon91088518 @starfess Ralat drak...	1774046917066133777	
179	1773337410912358426	Thu Mar 28 13:46:24	0	@starfess ini maksudnya sumintop marry my	1773345902146498927	

		+0000 2024		suffinten marry my husb...		
180	1776080598593134647	Fri Apr 05 02:55:34 +0000 2024	0	emang cuman di drama korea lelaki super green ...	1776081217290727451	

181 rows × 15 columns

```
# Check how many data we get

num_tweets = len(df)
print(f"Amount of tweets in dataframe is {num_tweets}.")
```

```
Amount of tweets in dataframe is 181.
```

```
new_df = df[['full_text']].copy()

# Display the DataFrame with only the 'full_text' column
new_df
```

	full_text
0	Beautiful Kang Yoomi #ParkMinYoung #MarryMyHus...
1	Kang Jiwon's cute paparazzi #유지혁 #강지원 #나인우 #박민...
2	@kdrama_menfess Yang jadi sumin di marry my hu...
3	Comel dahla matching color baju eee #marrymyhu...
4	@kdrama_menfess Aku prefer suminah (marry my h...
...	...
176	Terseret Skandal Kekerasan Foto SD Song Ha Yoo...
177	@Hsnmub @tang__kira Fight for My Way jadi teme...
178	@Soletil @Jaeyoon91088518 @starfess Ralat drak...
179	@starfess ini maksudnya suminten marry my husb...
180	emang cuman di drama korea lelaki super green ...

181 rows × 1 columns

Label Anotator Sentiment

To insert labels such as 'positive', 'neutral', or 'negative' for the DataFrame, we would typically go through a sentiment analysis process. This can be done either:

1. **Manually by reading through each text and assigning a label based on the sentiment conveyed**

For manual labeling, we could create a new column in the DataFrame and insert the labels directly.

2. **Automatically using a sentiment analysis tool or model**

For automatic labeling, we'd typically use a pre-trained sentiment analysis model from a library like NLTK, TextBlob, or through a service like Google Cloud Natural Language API.

Automatically with TextBlob

```
from textblob import TextBlob

auto_df = df[['full_text']].copy()
# Function to assign sentiment
def assign_sentiment(text):
    analysis = TextBlob(text)
    # Set our own thresholds for polarity
    if analysis.sentiment.polarity > 0.1:
        return 'positive'
    elif analysis.sentiment.polarity < -0.1:
        return 'negative'
    else:
        return 'neutral'

# Apply the function to the 'full_text' column
auto_df['sentiment'] = new_df['full_text'].apply(assign_sentiment)
auto_df
```

1 to 25 of 181 entries Filter

index	full_text	sentiment
0	Beautiful Kang Yoomi #ParkMinYoung #MarryMyHusband #Unstoppablehighkick ctto https://t.co/qMZVXyi4dC	positive
1	Kang Jiwon's cute paparazzi #유지혁 #강지원 #나인우 #박민영 #MarryMyHusband #내남결 #내남편과결혼해줘 https://t.co/v85nHupdtS	positive
2	@kdrama_menfess Yang jadi sumin di marry my husband ngga sih? Wah kalo bener sih harus di cancel tapi kalo ternyata cuman fitnah kasian baru aja naik namanya udh ada yang jahat nyebar rumor palsu.	neutral
3	Comel dahla matching color baju eee #marrymyhusband https://t.co/lg9mwp02B4	neutral
4	@kdrama_menfess Aku prefer suminah (marry my husband). Aktingnya ngena banget	neutral
5	Apakah ini efek dari Fanmeeting? Masa iya drama sudah selesai dari bulan lalu masih nongkrong di TOP 3 dan poinnya juga semakin bertambah? Terserah mau bilang apa tapi satu yang pasti pesona Marry My Husband memang nyata. Dan saya masih menikmatinya. #ParkMinYoung https://t.co/BFGmjwdn6g	positive
6	My 2Ji heart membuncah melimpah ruah #ParkMinYoung #박민영 #朴敏英 #パクミニョン #MYBRANDNEWDAYInJapan #MYBRANDNEWDAYInOsaka #MYbrandnewDay #NaInWoo #MarryMyHusband	neutral
7	And from now 'til my very last breath This day I'll cherish You look so beautiful in white.. Acaranya 2ji bulan Maret udh kelar nanti kita hajatan lagi tgl 20 april. Sehat2 terus ya kalian berdua #ParkMinYoung #박민영 #NaInWoo #나인우 #marrymyhusband https://t.co/CMaUQjNevW	positive
8	Terimakasih sudah memakai jasa kami Bagi yang mau order bisa langsung wa Untuk Tanya Price list harga!!! #zonauang #zonauang #eyeloveyou #fcklive #MarryMyHusband becak #хтивийпонеділок #jasahack #vcs #ajautr https://t.co/L9icE631k2	neutral
9	marry my husband seru juga tapi kalo keasikan nonton drakor gini kapan ngaji nya?	neutral
10	yang udah nonton drakor marry my husband seru ga?	neutral
11	CW KOREA // KDRAMA Ini jadinya gimana sih kelanjutan kasus bully villain marry my husband itu? Jadi dia bener pembully atau cuma dituduh aja? https://t.co/y9E52dC8n5	neutral
12	@adyppp Marry my husband top 1. Puas hati aku tgk dia revenge balik.	positive
13	LOH INI YG MAIN MARRY MY HUSBAND ITU KAN? BJIR KAGET BGT AING GA NGIKUTIN https://t.co/0XbzOLdy0h	positive
14	Kepincut abs nnton marry my husband. Senyumnya adem bener seperti mngajak berumah tangga haha	positive
15	mirip banget sama yg di marry my husband https://t.co/jrjd02Ju8T	neutral
16	marry my husband bukan sih judulnya wakakak maaf ga nonton dramanya	neutral
17	aaaaa telat bgt tp gpp sumpah marry my husband seruuuu bgttt !!! wajib nnton	neutral
18	@kdrama_menfess aduh mba su mi baru aja naik pamornya karna main di marry my husband	positive
19	kebayang victimnya ngeliat dia sukses di marry my husband pantes cocok bgt jd psikopat anjir 90 menit https://t.co/DsEeSA32B5	neutral
20	Sdh lbh dr sebulan tamat Marry my husband masih ranking 5 di weekday drama trend index ranking. https://t.co/tAk5pe8ZdK https://t.co/R9MuiKbdlr	neutral
21	marry my husband ni ada dua sebab tak boleh feel 1. tak suka park min young 2. sebab dah tengok na in woo kat 2d1n	neutral
22	Semoga casts #MarryMyHusband eligible buat masuk ke nominasi dan bisa borong piala taun ini	neutral
23	bentar ini tuh yg di drama marry my husband bukan?	neutral
24	baru tgk Marry my husband lah ada Gikwang ??	neutral

Show 25 per page

1 2 3 4 5 6 7 8

Like what you see? Visit the data table notebook to learn more about interactive tables.

Next steps: View recommended plots

Text Processing

Langkah-langkah yang digunakan dalam melakukan text preprocessing adalah sebagai berikut:

- Cleaning text
- Lowercase
- Remove stopwords
- Stemming / Lemmatization
- Tokenization

Cleaning Text and Lowercase

(remove urls, hashtags, mention (@), emojis, punctuations, and whitespaces)

```
def cleaning_text(text):
    # remove url
    url_pattern = re.compile(r'https?://\S+|www\.\S+')
    text = url_pattern.sub(r'', text)

    # remove hashtags
    # only removing the hash # sign from the word
    text = re.sub(r'#', '', text)

    # remove mention handle user (@)
    text = re.sub(r'@[\w]*', ' ', text)

    # remove emojis
    emoji_pattern = re.compile(
        '['
        '\U0001F600-\U0001F64F'  # emoticons
        '\U0001F300-\U0001F5FF'  # symbols & pictographs
        '\U0001F680-\U0001F6FF'  # transport & map symbols
        '\U0001F700-\U0001F77F'  # alchemical symbols
        '\U0001F780-\U0001F7FF'  # Geometric Shapes Extended
        '\U0001F800-\U0001F8FF'  # Supplemental Arrows-C
        '\U0001F900-\U0001F9FF'  # Supplemental Symbols and Pictographs
        '\U0001FA00-\U0001FA6F'  # Chess Symbols
        '\U0001FA70-\U0001FAFF'  # Symbols and Pictographs Extended-A
        '\U00002702-\U000027B0'  # Dingbats
        '\U000024C2-\U0001F251'
        ']+',
        flags=re.UNICODE
    )
    text = emoji_pattern.sub(r'', text)

    # remove punctuation
    punctuations = '''!()-[]{};:'"\,<>./?@#$%^&*_~'''
    for x in text.lower():
        if x in punctuations:
            text = text.replace(x, " ")

```

```
    # remove extra whitespace
    text = ' '.join(text.split())

    # lowercase
    text = text.lower()
    return text
```

Remove Stopword

```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
nltk.download('punkt')

# CONSTRUCT STOPWORDS
rama_stopword = "https://raw.githubusercontent.com/ramaprakoso/analisis-sentimen/master/kamus/stopword.txt"
yutomo_stopword = "https://raw.githubusercontent.com/yasirutomo/python-sentianalysis-id/master/data/feature_list/stopwordsID.txt
fpmipa_stopword = "https://raw.githubusercontent.com/onlyphantom/elangdev/master/elang/word2vec/utils/stopwords-list/fpmipa-stop
sastrawi_stopword = "https://raw.githubusercontent.com/onlyphantom/elangdev/master/elang/word2vec/utils/stopwords-list/sastrawi-
aliakbar_stopword = "https://raw.githubusercontent.com/onlyphantom/elangdev/master/elang/word2vec/utils/stopwords-list/aliakbars
pebahasa_stopword = "https://raw.githubusercontent.com/onlyphantom/elangdev/master/elang/word2vec/utils/stopwords-list/pebbie-pe
elang_stopword = "https://raw.githubusercontent.com/onlyphantom/elangdev/master/elang/word2vec/utils/stopwords-id.txt"
nltk_stopword = stopwords.words('indonesian')

# create path url for each stopword
path_stopwords = [rama_stopword, yutomo_stopword, fpmipa_stopword, sastrawi_stopword,
                  aliakbar_stopword, pebahasa_stopword, elang_stopword]

# combine stopwords
stopwords_l = nltk_stopword
for path in path_stopwords:
    response = requests.get(path)
    stopwords_l += response.text.split('\n')

custom_st = '''
yg yang dgn ane smpai bgt ga gk nggak gua gwa gw si tu ama utk tuk udh dah btw
ntar entar lol ttg emg aj aja tll sy sih kalo nya trsa mnrt nih tgk
ma dr ajaa tp akan bs bikin kta pas pdahl bnyak guys abis tnx
bang banget nang mas amat bangettt tjoy hemm haha sllu hrs lanjut
bgtu sbnrnya trjadi bgtu pdhl sm plg skrg km njir bjir cuman
'''

# create dictionary with unique stopword
st_words = set(stopwords_l)
custom_stopword = set(custom_st.split())

```

```
# result stopwords
stop_words = st_words | custom_stopword
print(f'Stopwords: {list(stop_words)[:5]}')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
Stopwords: ['manalagi', 'sebagaimana', 'agar', 'panjang', 'the']
```

```
# remove stopwords
from nltk import word_tokenize, sent_tokenize

def remove_stopword(text, stop_words=stop_words):
    word_tokens = word_tokenize(text)
    filtered_sentence = [w for w in word_tokens if not w in stop_words]
    return ' '.join(filtered_sentence)
```

Stemming / Lemmatization

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

```
# stemming and lemmatization

def stemming_and_lemmatization(text):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    return stemmer.stem(text)
```

Tokenization

```
# tokenization
def tokenize(text):
    return word_tokenize(text)
```

Try Function in text processing

```
# example
text = '''
Untungnya agensi langsung ambil tindakan dengan membantah, Bun! Bahkan ada teman Song Ha Yoon yang angkat suara mengenai kasus i
Cek artikel lainnya> https://haibunda.com/trending/20240403081248-93-333499/kata-agensi-soal-kabar-song-ha-yoon-bintang-marry-my
'''
print(f'Original text: \n{text}\n')

# cleaning text and lowercase
text = cleaning_text(text)
print(f'Cleaned text: \n{text}\n')

# remove stopwords
text = remove_stopword(text)
print(f'Removed stopword: \n{text}\n')

# stemming and lemmatization
text = stemming_and_lemmatization(text)
print(f'Stemmed and lemmatized: \n{text}\n')

# tokenization
text = tokenize(text)
print(f'Tokenized: \n{text}')
```

```
Original text:

Untungnya agensi langsung ambil tindakan dengan membantah, Bun! Bahkan ada teman Song Ha Yoon yang angkat suara mengenai kasus
Cek artikel lainnya> https://haibunda.com/trending/20240403081248-93-333499/kata-agensi-soal-kabar-song-ha-yoon-bintang-marry-m
```

```
Cleaned text:
untungnya agensi langsung ambil tindakan dengan membantah bun bahkan ada teman song ha yoon yang angkat suara mengenai kasus in

Removed stopword:
untungnya agensi ambil tindakan membantah bun song ha yoon angkat suara duh semoga cepat selesai bun songhayoon marrymyhusband

Stemmed and lemmatized:
untung agens ambil tindak ban bun song ha yoon angkat suara duh moga cepat selesai bun songhayoon marrymyhusband cek artikel

Tokenized:
['untung', 'agens', 'ambil', 'tindak', 'ban', 'bun', 'song', 'ha', 'yoon', 'angkat', 'suara', 'duh', 'moga', 'cepat', 'selesai'
```

Implement to our dataset

```
# pipeline preprocess
def preprocess(text):
    # cleaning text and lowercase
    output = cleaning_text(text)

    # remove stopwords
    output = remove_stopword(output)

    return output
```

```
# implement preprocessing
preprocessed_data = auto_df.copy()
preprocessed_data['full_text'] = auto_df['full_text'].map(preprocess)
```

```
preprocessed_data.head()
```

1 to 5 of 5 entries Filter

index	full_text	sentiment
0	beautiful kang yoomi parkminyoung marrymyhusband unstoppablehighkick ctto	positive
1	kang jiwon s cute paparazzi marrymyhusband	positive
2	sumin marry my husband ngga bener cancel fitnah kasian namanya jahat nyebar rumor palsu	neutral
3	comel dahla matching color baju eee marrymyhusband	neutral
4	prefer suminah marry my husband aktingnya ngena	neutral

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Next steps: View recommended plots

```
1 preprocessed_data['full_text'][180]
```

```
'drama korea lelaki super green flag tuh nonton marry my husband geregetan emosi ngakak grgr gwenchana role jd antagonis'
```

Save to CSV

```
1 df = pd.DataFrame(preprocessed_data)
2 csv_file_path = 'with_preprocessed_data.csv'
3 df.to_csv(csv_file_path, sep=';', index=False, header=True)
4
5 print(f'Data has been saved to {csv_file_path}')
```

```
Data has been saved to with_preprocessed_data.csv
```

```
# load dataset into pandas
import pandas as pd
data = pd.read_csv('/content/with_preprocessed_data.csv', sep=';')
data
```

1 to 25 of 181 entries Filter

index	full_text	sentiment
0	beautiful kang yoomi parkminyoung marrymyhusband unstoppablehighkick ctto	positive
1	kang jiwon s cute paparazzi marrymyhusband	positive
2	sumin marry my husband ngga bener cancel fitnah kasian namanya jahat nyebar rumor palsu	neutral
3	comel dahla matching color baju eee marrymyhusband	neutral
4	prefer suminah marry my husband aktingnya ngena	neutral
5	efek fanmeeting drama selesai nongkrong top 3 poinnya bertambah terserah bilang pesona marry my husband nyata menikmatinya parkminyoung	positive
6	my 2ji heart membuncah melimpah ruah parkminyoung mybrandnewdayinjapan mybrandnewdayinosaka mybrandnewday nainwoo marrymyhusband	neutral
7	and from now til my very last breath this day i ll cherish you look so beautiful in white acaranya 2ji maret kelar hajatan tgl 20 april sehat2 berdua parkminyoung nainwoo marrymyhusband	positive
8	terimakasih memakai jasa order wa price list harga zonauang zonauang eyeloveyou fcklive marrymyhusband becak хтивийпонеділок jasahack vcs ajautr	neutral
9	marry my husband seru keasikan nonton drakor ngaji	neutral
10	nonton drakor marry my husband seru	neutral
11	cw korea kdrama kelanjutan bully villain marry my husband bener pembully dituduh	neutral
12	marry my husband top 1 puas hati revenge	positive
13	main marry my husband kaget aing ngikutin	positive
14	kepincut abs nnton marry my husband senyumnya adem bener mngajak berumah tangga	positive
15	marry my husband	neutral
16	marry my husband judulnya wakakak maaf nonton dramanya	neutral
17	aaaaa telat gpp sumpah marry my husband seruuuu bgttt wajib nnton	neutral
18	aduh su mi pamornya karna main marry my husband	positive
19	kebayang victimnya ngeliat sukses marry my husband pantes cocok jd psikopat anjir 90 menit	neutral
20	sdh lbh sebulan tamat marry my husband ranking 5 weekday drama trend index ranking	neutral
21	marry my husband ni feel 1 suka park min young 2 tengok na in woo kat 2d1n	neutral
22	semoga casts marrymyhusband eligible nominasi borong piala taun	neutral
23	bentar tuh drama marry my husband	neutral
24	marry my husband gikwang	neutral

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Next steps: View recommended plots

Build LSTM and Model Training

```
import pandas as pd
from sklearn.model_selection import train_test_split
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Embedding, LSTM, Dense

# Assuming your data is in a CSV file
file_path = '/content/with_preprocessed_data.csv'
df = pd.read_csv(file_path, sep=';')

# Assuming the 'Text Tweet' column contains the text data and 'Sentiment' contains labels
texts = df['full_text'].tolist()
labels = df['sentiment'].tolist()

# Tokenize the text data
max_words = 10000  # Adjust based on our dataset size
tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(texts)
sequences = tokenizer.texts_to_sequences(texts)

# Pad sequences to make them of equal length
max_sequence_length = 100  # Adjust based on our dataset and sequence length
data = pad_sequences(sequences, maxlen=max_sequence_length)

# Convert labels to one-hot encoding
labels = pd.get_dummies(labels)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)

# Build the LSTM model
model = Sequential()
model.add(Embedding(input_dim=max_words, output_dim=100, input_length=max_sequence_length))
model.add(LSTM(units=64, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(units=len(labels.columns), activation='softmax'))

```

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=300, batch_size=32, validation_data=(X_test, y_test))

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test Loss: {loss:.4f}, Test Accuracy: {accuracy:.4f}")
```

```
5/5 [==============================] - 1s 164ms/step - loss: 7.5858e-05 - accuracy: 1.0000 - val_loss: 1.5322 - val_accuracy
Epoch 290/300
5/5 [==============================] - 1s 168ms/step - loss: 7.6886e-05 - accuracy: 1.0000 - val_loss: 1.5317 - val_accuracy
Epoch 291/300
5/5 [==============================] - 1s 162ms/step - loss: 7.4767e-05 - accuracy: 1.0000 - val_loss: 1.5353 - val_accuracy
Epoch 292/300
5/5 [==============================] - 1s 157ms/step - loss: 7.7982e-05 - accuracy: 1.0000 - val_loss: 1.5399 - val_accuracy
Epoch 293/300
5/5 [==============================] - 1s 158ms/step - loss: 8.7532e-05 - accuracy: 1.0000 - val_loss: 1.5435 - val_accuracy
Epoch 294/300
5/5 [==============================] - 1s 166ms/step - loss: 8.5994e-05 - accuracy: 1.0000 - val_loss: 1.5425 - val_accuracy
Epoch 295/300
5/5 [==============================] - 1s 190ms/step - loss: 7.6094e-05 - accuracy: 1.0000 - val_loss: 1.5450 - val_accuracy
Epoch 296/300
5/5 [==============================] - 1s 267ms/step - loss: 9.3870e-05 - accuracy: 1.0000 - val_loss: 1.5484 - val_accuracy
Epoch 297/300
5/5 [==============================] - 1s 291ms/step - loss: 8.5721e-05 - accuracy: 1.0000 - val_loss: 1.5481 - val_accuracy
Epoch 298/300
5/5 [==============================] - 1s 204ms/step - loss: 7.2228e-05 - accuracy: 1.0000 - val_loss: 1.5568 - val_accuracy
Epoch 299/300
5/5 [==============================] - 1s 155ms/step - loss: 7.1537e-05 - accuracy: 1.0000 - val_loss: 1.5632 - val_accuracy
Epoch 300/300
5/5 [==============================] - 1s 160ms/step - loss: 1.2164e-04 - accuracy: 1.0000 - val_loss: 1.5511 - val_accuracy
2/2 [==============================] - 0s 23ms/step - loss: 1.5511 - accuracy: 0.8108
Test Loss: 1.5511, Test Accuracy: 0.8108
```

Evaluate Model

```
from sklearn.metrics import classification_report, accuracy_score, f1_score, roc_auc_score
from tensorflow.keras.utils import to_categorical
import numpy as np
```

```
1 # Predict the probabilities for each class
```