Source code:

from textblob import TextBlob

import pandas as pd

# Creating an example DataFrame. Replace with your data loading logic.

df = pd.DataFrame({'clean\_text': ['This is a good example', 'This is a bad example', 'This is neutral']})

def get\_sentiment(text):

polarity = TextBlob(text).sentiment.polarity

if polarity > 0:

return 'Positive'

elif polarity == 0:

return 'Neutral'

else:

return 'Negative'

df['sentiment'] = df['clean\_text'].apply(get\_sentiment)

print(df['sentiment'].value\_counts())

import pandas as pd

# Sample Instagram comments

comments = [

"I love this picture so much!",

"This is horrible!",

"What an amazing day!",

"I'm so sad right now.",

"Congratulations, you deserve it!",

"Feeling angry about this.",

"So excited to see this!",

"I feel scared about the future.",

"Absolutely wonderful work!",

"Disappointed by this event."

]

# Create a dataframe

df = pd.DataFrame(comments, columns=['text'])

df.to\_csv('instagram\_comments.csv', index=False)

print("Sample comments saved!")

import re

import nltk

from nltk.corpus import stopwords

import pandas as pd # Import pandas here

import os # import the os module

# Download the Punkt sentence tokenizer data

nltk.download('punkt\_tab') # this line was added to fix the error

nltk.download('stopwords')

nltk.download('punkt')

def clean\_text(text):

text = text.lower()

text = re.sub(r'@\w+','',text) # Remove mentions

text = re.sub(r'http\S+','',text) # Remove links

text = re.sub(r'[^a-zA-Z\s]', '', text) # Remove punctuation and emojis

tokens = nltk.word\_tokenize(text)

tokens = [word for word in tokens if word not in stopwords.words('english')]

return ' '.join(tokens)

# Get the current working directory

current\_directory = os.getcwd()

print(f"Current working directory: {current\_directory}")

# Construct the full path to the CSV file

csv\_file\_path = os.path.join(current\_directory, 'instagram\_comments.csv')

# Check if the file exists

if os.path.exists(csv\_file\_path):

# Load the DataFrame from the CSV file

df = pd.read\_csv(csv\_file\_path) # Load df from the saved CSV

df['clean\_text'] = df['text'].apply(clean\_text) # Now df is defined in this cell

else:

print(f"Error: The file '{csv\_file\_path}' does not exist.")

def detect\_emotion(text):

emotions = {

'happy': ['happy', 'joy', 'excited', 'love', 'wonderful'],

'sad': ['sad', 'depressed', 'cry', 'unhappy'],

'angry': ['angry', 'mad', 'furious'],

'surprise': ['surprised', 'amazed', 'shocked'],

'fear': ['fear', 'scared', 'afraid']

}

tokens = text.split()

for emotion, keywords in emotions.items():

if any(word in tokens for word in keywords):

return emotion

return 'neutral'

df['emotion'] = df['clean\_text'].apply(detect\_emotion)

print(df['emotion'].value\_counts())

import matplotlib.pyplot as plt

import seaborn as sns

from wordcloud import WordCloud

import pandas as pd

import re

import nltk

from nltk.corpus import stopwords

import os

# Assuming 'instagram\_comments.csv' is in the same directory

df = pd.read\_csv('instagram\_comments.csv')

# ... (rest of the code for cleaning and processing the dataframe as before) ...

# Download the Punkt sentence tokenizer data

nltk.download('punkt\_tab') # this line was added to fix the error

nltk.download('stopwords')

nltk.download('punkt')

def clean\_text(text):

text = text.lower()

text = re.sub(r'@\w+','',text) # Remove mentions

text = re.sub(r'http\S+','',text) # Remove links

text = re.sub(r'[^a-zA-Z\s]', '', text) # Remove punctuation and emojis

tokens = nltk.word\_tokenize(text)

tokens = [word for word in tokens if word not in stopwords.words('english')]

return ' '.join(tokens)

df['clean\_text'] = df['text'].apply(clean\_text) # Now df is defined in this cell

from textblob import TextBlob # Import TextBlob

def get\_sentiment(text):

polarity = TextBlob(text).sentiment.polarity

if polarity > 0:

return 'Positive'

elif polarity == 0:

return 'Neutral'

else:

return 'Negative'

df['sentiment'] = df['clean\_text'].apply(get\_sentiment) # Calculate sentiment

def detect\_emotion(text):

emotions = {

'happy': ['happy', 'joy', 'excited', 'love', 'wonderful'],

'sad': ['sad', 'depressed', 'cry', 'unhappy'],

'angry': ['angry', 'mad', 'furious'],

'surprise': ['surprised', 'amazed', 'shocked'],

'fear': ['fear', 'scared', 'afraid']

}

tokens = text.split()

for emotion, keywords in emotions.items():

if any(word in tokens for word in keywords):

return emotion

return 'neutral'

df['emotion'] = df['clean\_text'].apply(detect\_emotion) # Calculate emotion

# Sentiment Plot

plt.figure(figsize=(6,4))

sns.countplot(x='sentiment', data=df, palette='coolwarm') # Now 'sentiment' column is available

plt.title('Sentiment Distribution')

plt.show()

# Emotion Plot

plt.figure(figsize=(6,4))

sns.countplot(x='emotion', data=df, palette='viridis') # Now 'emotion' column is available

plt.title('Emotion Distribution')

plt.show()

OUTPUT:











