Sentiment analysis, also known as opinion mining, is the process of using␣

- natural language processing (NLP) techniques to determine the sentiment or emotional tone expressed in text data. When applied to social media data, sentiment analysis can provide valuable
- insights into public opinion, customer feedback, brand perception, and more. Here's a brief overview of sentiment analysis using␣
- social media data:

Use a dataset of tweets or Facebook posts and perform sentiment analysis to determine the overall sentiment of the posts.

panda,numpy,matplotlib,seaborn,sklearn are the basic libraries used in the email spam filtering natural language tool kit used to study the data which means a mail and visualized the data in the different graphical form(pictorial representation

Packages requried for the analysis

• nltk: natural language tool kit used for text analysis

• pandas : used for anlayse dataframe

• matplotlib and seborn: used for plotting

In [3]:

```
pip install vadersentiment
```

```
Requirement already satisfied: vadersentiment in c:\users\prade\anaconda3\lib\site-packages (3.3.2)
Requirement already satisfied: requests in c:\users\prade\anaconda3\lib\site-packages (from vadersentiment) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\prade\anaconda3\lib\site-packages (from requests->vad
ersentiment) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\prade\anaconda3\lib\site-packages (from requests->vadersentiment)
(3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\prade\anaconda3\lib\site-packages (from requests->vadersent
iment) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\prade\anaconda3\lib\site-packages (from requests->vadersent
iment) (2023.7.22)
Note: you may need to restart the kernel to use updated packages.
```

In [1]:

```
pip install wordcloud
```

```
Collecting wordcloudNote: you may need to restart the kernel to use updated packages.

  Obtaining dependency information for wordcloud from https://files.pythonhosted.org/packages/f5/b0/247159f61c5d5d6647171
bef84430b7efad4db504f0229674024f3a4f7f2/wordcloud-1.9.3-cp311-cp311-win_amd64.whl.metadata (https://files.pythonhosted.or
g/packages/f5/b0/247159f61c5d5d6647171bef84430b7efad4db504f0229674024f3a4f7f2/wordcloud-1.9.3-cp311-cp311-win_amd64.whl.m
etadata)
  Downloading wordcloud-1.9.3-cp311-cp311-win_amd64.whl.metadata (3.5 kB)
Requirement already satisfied: numpy>=1.6.1 in c:\users\prade\anaconda3\lib\site-packages (from wordcloud) (1.24.3)
Requirement already satisfied: pillow in c:\users\prade\anaconda3\lib\site-packages (from wordcloud) (9.4.0)
Requirement already satisfied: matplotlib in c:\users\prade\anaconda3\lib\site-packages (from wordcloud) (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\prade\anaconda3\lib\site-packages (from matplotlib->wordclou
d) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\prade\anaconda3\lib\site-packages (from matplotlib->wordcloud)
(0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\prade\anaconda3\lib\site-packages (from matplotlib->wordclou
d) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\prade\anaconda3\lib\site-packages (from matplotlib->wordclou
d) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\prade\anaconda3\lib\site-packages (from matplotlib->wordcloud)
(23.1)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\prade\anaconda3\lib\site-packages (from matplotlib->word
cloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\prade\anaconda3\lib\site-packages (from matplotlib->wordc
loud) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\prade\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplot
lib->wordcloud) (1.16.0)
Downloading wordcloud-1.9.3-cp311-cp311-win_amd64.whl (300 kB)
   ---------------------------------------- 0.0/300.2 kB ? eta -:--:--
   ---------------------------------------- 300.2/300.2 kB 9.4 MB/s eta 0:00:00
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.3
```

In [2]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.corpus import stopwords
from nltk import PorterStemmer
from wordcloud import WordCloud
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

Reading the data

In [3]:
```
df=pd.read_csv("C:\\Users\\prade\\OneDrive\\Documents\\DATASCIENCE\\Intern DataSets\\archive\\Tweets.csv")
df
```

Out[3]:

onfidence	negativereason	negativereason_confidence	airline	airline_sentiment_gold	name	negativereason_gold	retweet_count	te
1.0000	NaN	NaN	Virgin America	NaN	cairdin	NaN	0	@VirginAmeri Wh @dhepbu sai
0.3486	NaN	0.0000	Virgin America	NaN	jnardino	NaN	0	@VirginAmeri plus you' add commercials t
0.6837	NaN	NaN	Virgin America	NaN	yvonnalynn	NaN	0	@VirginAmeri I didn't today Must mean I n
1.0000	Bad Flight	0.7033	Virgin America	NaN	jnardino	NaN	0	@VirginAmeri it's rea aggressive blast
1.0000	Can't Tell	1.0000	Virgin America	NaN	jnardino	NaN	0	@VirginAmeri and it's a rea big bad thing
...	...	...	...	...	...	...	...	
0.3487	NaN	0.0000	American	NaN	KristenReenders	NaN	0	@American thank you v got on different f
1.0000	Customer Service Issue	1.0000	American	NaN	itsropes	NaN	0	@American leaving over 2 minutes La Flig
1.0000	NaN	NaN	American	NaN	sanyabun	NaN	0	@American Please bri America Airlines to
1.0000	Customer Service Issue	0.6659	American	NaN	SraJackson	NaN	0	@American you have n money, y change my
0.6771	NaN	0.0000	American	NaN	daviddtwu	NaN	0	@American we have 8 p so we need know h

In [4]:
```
df.columns
```

Out[4]:
```
Index(['tweet_id', 'airline_sentiment', 'airline_sentiment_confidence',
       'negativereason', 'negativereason_confidence', 'airline',
       'airline_sentiment_gold', 'name', 'negativereason_gold',
       'retweet_count', 'text', 'tweet_coord', 'tweet_created',
       'tweet_location', 'user_timezone'],
      dtype='object')
```

In [6]:
```
df.shape
```

Out[6]:
```
(14640, 15)
```

In [7]:
```
df.size
```

Out[7]:
```
219600
```

In [9]:
```
df.dtypes
```

Out[9]:
```
tweet_id                          int64
airline_sentiment                object
airline_sentiment_confidence    float64
negativereason                   object
negativereason_confidence       float64
airline                          object
airline_sentiment_gold           object
name                             object
negativereason_gold              object
retweet_count                     int64
text                             object
tweet_coord                      object
tweet_created                    object
tweet_location                   object
user_timezone                    object
dtype: object
```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
tweet_id                            0
airline_sentiment                   0
airline_sentiment_confidence        0
negativereason                   5462
negativereason_confidence        4118
airline                             0
airline_sentiment_gold          14600
name                                0
negativereason_gold             14608
retweet_count                       0
text                                0
tweet_coord                     13621
tweet_created                       0
tweet_location                   4733
user_timezone                    4820
dtype: int64
```

In [11]:
```
df=df.dropna()
```

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
tweet_id                        0
airline_sentiment               0
airline_sentiment_confidence    0
negativereason                  0
negativereason_confidence       0
airline                         0
airline_sentiment_gold          0
name                            0
negativereason_gold             0
retweet_count                   0
text                            0
tweet_coord                     0
tweet_created                   0
tweet_location                  0
user_timezone                   0
dtype: int64
```

In [13]:
```
display(df.shape)
display(df.info())
```

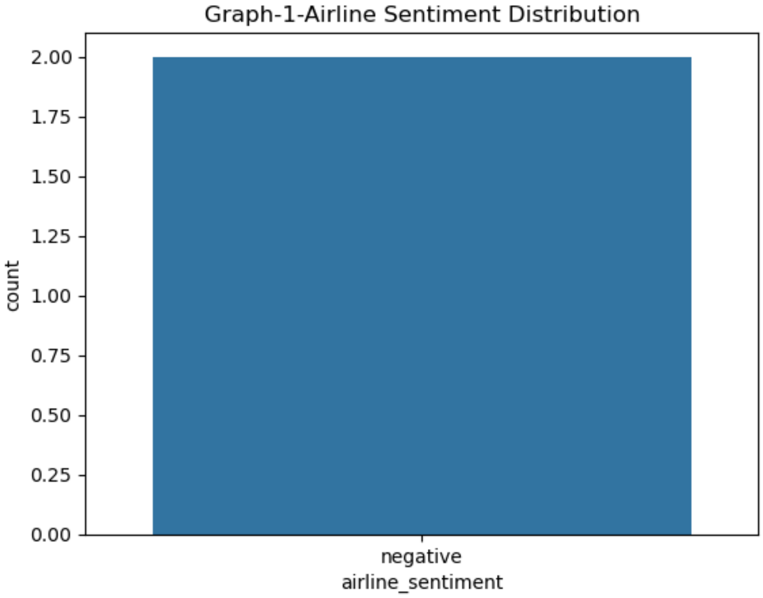
```
(2, 15)

<class 'pandas.core.frame.DataFrame'>
Index: 2 entries, 4206 to 9536
Data columns (total 15 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   tweet_id                      2 non-null      int64  
 1   airline_sentiment             2 non-null      object 
 2   airline_sentiment_confidence  2 non-null      float64
 3   negativereason                2 non-null      object 
 4   negativereason_confidence     2 non-null      float64
 5   airline                       2 non-null      object 
 6   airline_sentiment_gold        2 non-null      object 
 7   name                          2 non-null      object 
 8   negativereason_gold           2 non-null      object 
 9   retweet_count                 2 non-null      int64  
 10  text                          2 non-null      object 
 11  tweet_coord                   2 non-null      object 
 12  tweet_created                 2 non-null      object 
 13  tweet_location                2 non-null      object 
 14  user_timezone                 2 non-null      object 
dtypes: float64(2), int64(2), object(11)
memory usage: 256.0+ bytes

None
```

In [14]:
```
#redefining dataset for analysis
df=df[['airline_sentiment','text']]
df
```

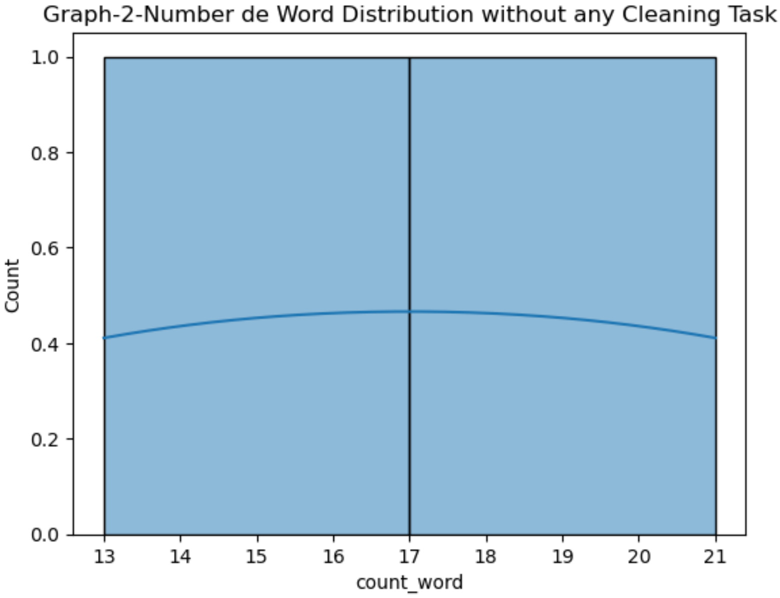
Out[14]:

	airline_sentiment	text
4206	negative	@united So what do you offer now that my fligh...
9536	negative	@USAirways Seriously doubt that as I am still ...

```
In [15]: # airline_sentiment distributionn
         sns.countplot(data=df,x='airline_sentiment')
         plt.title('Graph-1-Airline Sentiment Distribution')
         plt.show()
```

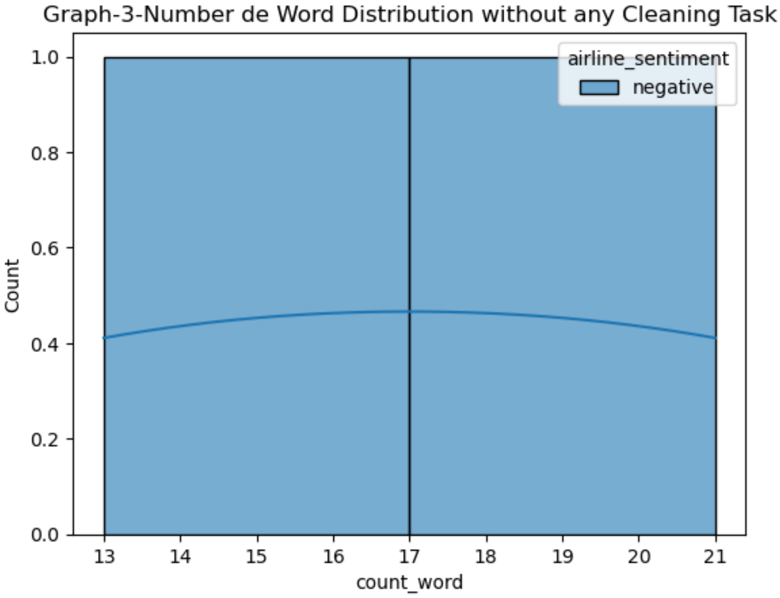


```
In [16]: # creating a new column counting the number of word in each tweets

         df['count_word'] = df['text'].apply(lambda x : len(x.split(' ')))
         sns.histplot(data = df , x='count_word',kde=True)
         plt.title('Graph-2-Number de Word Distribution without any Cleaning Task')
         plt.show()
```



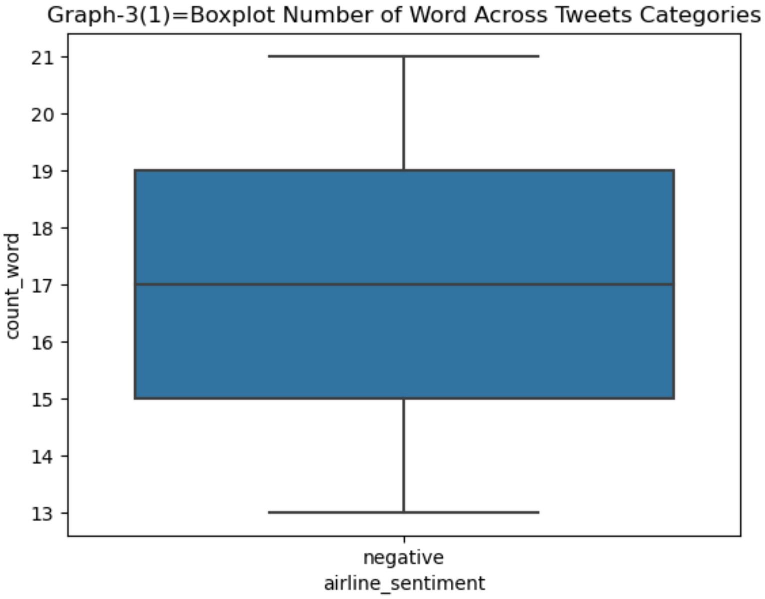
```
In [17]: #word distribution---without cleaning the data
```

In [18]:
```
sns.histplot(data = df , x='count_word',hue='airline_sentiment',alpha=0.6,
             kde=True)
plt.title('Graph-3-Number de Word Distribution without any Cleaning Task')
plt.show()
```



In [19]:
```
#using the box plots to visulaize the words at tweets more better.
```

In [20]:
```
sns.boxplot(data = df , y='count_word',x='airline_sentiment')
plt.title('Graph-3(1)=Boxplot Number of Word Across Tweets Categories')
plt.show()
```



In [21]:
```
df.loc[np.logical_or(df['count_word']>35,df['count_word']<=5),:]
```

Out[21]:

airline_sentiment	text	count_word

In [22]:
```
# Preprocessing the data:

# Punctuation Removal

# StopWord Removal

# Numeric Values Removal

# Stemming

# Tokenization
```

In [23]:
```
# import preprocessing libraries
import re
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
```

In [24]:
```
# punctuation Removal
def remove_punctuation(text):
    return re.sub(r'[^\w\s]','',text)
```

In [25]:
```
#stopword removal
def remove_stopwords(text):
    stop_words = set(stopwords.words('english'))
    tokens = word_tokenize(text)
    filter_tokens = [word for word in tokens if word.lower() not in stop_words]
    return " ".join(filter_tokens)
```

In [26]:
```
#remove numeric
def remove_numeric(text):
    return re.sub(r'\d+','',text)
```

In [27]:
```
#Stemming
def apply_stemming(text):
    stemmer = PorterStemmer()
    tokens = word_tokenize(text)
    stemmed_tokens = [stemmer.stem(word) for word in tokens]
    return " ".join(stemmed_tokens)
```

In [28]:
```
def remove_mentions(text):
    return re.sub(r'@\w+','',text)
```

In [29]:
```
import nltk
nltk.download('punkt')

from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\prade\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

In [30]:
```
def apply_stemming(text):
    stemmer = PorterStemmer()
    tokens = word_tokenize(text)
    stemmed_tokens = [stemmer.stem(word) for word in tokens]
    return " ".join(stemmed_tokens)
```

In [31]:
```
input_text = "walking throw the street, a passenger walked toward me, talking about a walked chicken on the streets"
stemmed_text = apply_stemming(input_text)
print(stemmed_text)
```

```
walk throw the street , a passeng walk toward me , talk about a walk chicken on the street
```

In [32]:
```
# sample stemming
apply_stemming('walking throw the street , a passenger walked toward me,talking about a walked chicken on the streets')
```

Out[32]:
```
'walk throw the street , a passeng walk toward me , talk about a walk chicken on the street'
```

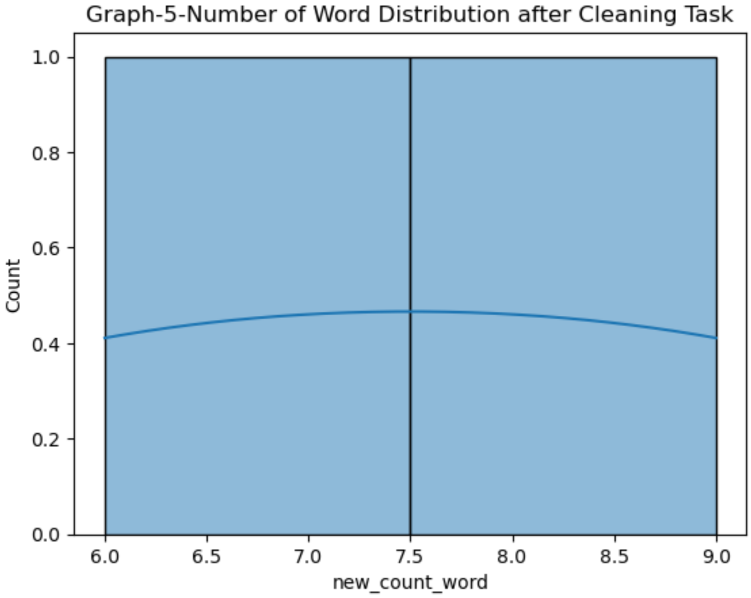
In [33]:
```
# General Preprocessing Function
def text_preprocessing(text):
    sentence = remove_mentions(text)
    sentence = remove_punctuation(sentence)
    sentence = remove_stopwords(sentence)
    sentence = remove_numeric(sentence)
    sentence = apply_stemming(sentence)
    return sentence
```

In [34]:
```
text_preprocessing('walking throw the street , a passenger walked toward me,talking about a walked chicken on the streets'
```

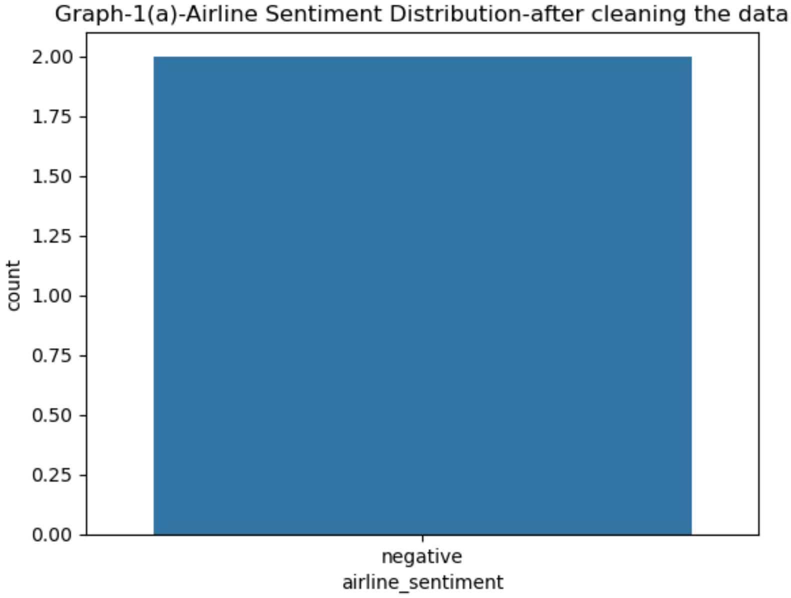
Out[34]:
```
'walk throw street passeng walk toward metalk walk chicken street'
```

In [35]:
```
df.loc[:,'new_text'] = df['text'].apply(lambda x : text_preprocessing(x))
```

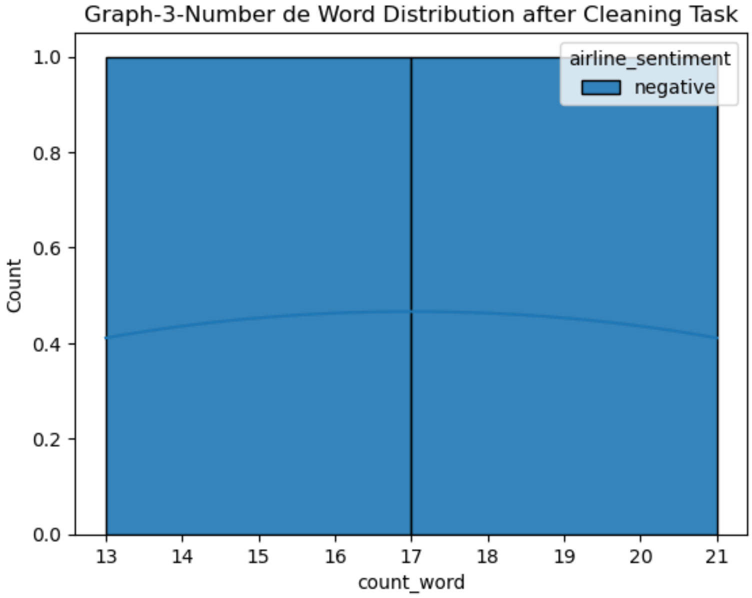
In [36]:
```
df.loc[:,'new_count_word'] = df['new_text'].apply(lambda x : len(x.split(' ')))
sns.histplot(data = df , x='new_count_word',kde=True)
plt.title('Graph-5-Number of Word Distribution after Cleaning Task')
plt.show()
```



In [37]:
```
# airline_sentiment distributionn
sns.countplot(data=df,x='airline_sentiment')
plt.title('Graph-1(a)-Airline Sentiment Distribution-after cleaning the data')
plt.show()
```



In [38]:
```
sns.histplot(data = df , x='count_word',hue='airline_sentiment',alpha=0.9,
             kde=True)
plt.title('Graph-3-Number de Word Distribution after Cleaning Task')
plt.show()
```



In [ ]: