import required libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
import nltk
from nltk.corpus import stopwords
import re
import joblib
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
```

Filter warnings

In [3]:
```
import warnings
warnings.filterwarnings("ignore")
```

In [4]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\hi\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[4]: True

adding columns names

In [5]:
```
columns = ['sentiment', 'ids', 'date', 'flag', 'user', 'text']
```

Loading dataset

In [6]:
```
data=pd.read_csv("training.1600000.processed.noemoticon.csv",encoding="ISO-8859-
```

Access first five rows

In [7]:
```
data.head()
```

Out[7]:

	sentiment	ids	date	flag	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

Access last five rows

In [8]:
```
data.tail()
```

Out[8]:

	sentiment	ids	date	flag	user	te
1599995	4	2193601966	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	AmandaMarie1028	Just woke u Having r school is tl best fee
1599996	4	2193601969	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	TheWDBoards	TheWDB.com Very cool to he old Walt interv
1599997	4	2193601991	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	bpbabe	Are you rea for your Mo Makeover? A me
1599998	4	2193602064	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	tinydiamondz	Happy 38 Birthday to n boo of alll time
1599999	4	2193602129	Tue Jun 16 08:40:50 PDT 2009	NO_QUERY	RyanTrevMorris	hapı #charitytuesd @theNSP(@SparksCharity

identify no.of columns and rows

In [9]:
```
data.shape
```

Out[9]:
```
(1600000, 6)
```

Dispaly column names

In [10]:
```
data.columns
```

Out[10]:
```
Index(['sentiment', 'ids', 'date', 'flag', 'user', 'text'], dtype='object')
```

Access target column and unqiue values

In [11]:
```
data.sentiment.unique()
```

Out[11]:
```
array([0, 4])
```

replace (4,0) to (1,0)

In [12]:
```
data["sentiment"] = data.sentiment.replace((4,0),(1,0))
```

In [13]:
```
data.sentiment.unique()
```

Out[13]:
```
array([0, 1])
```

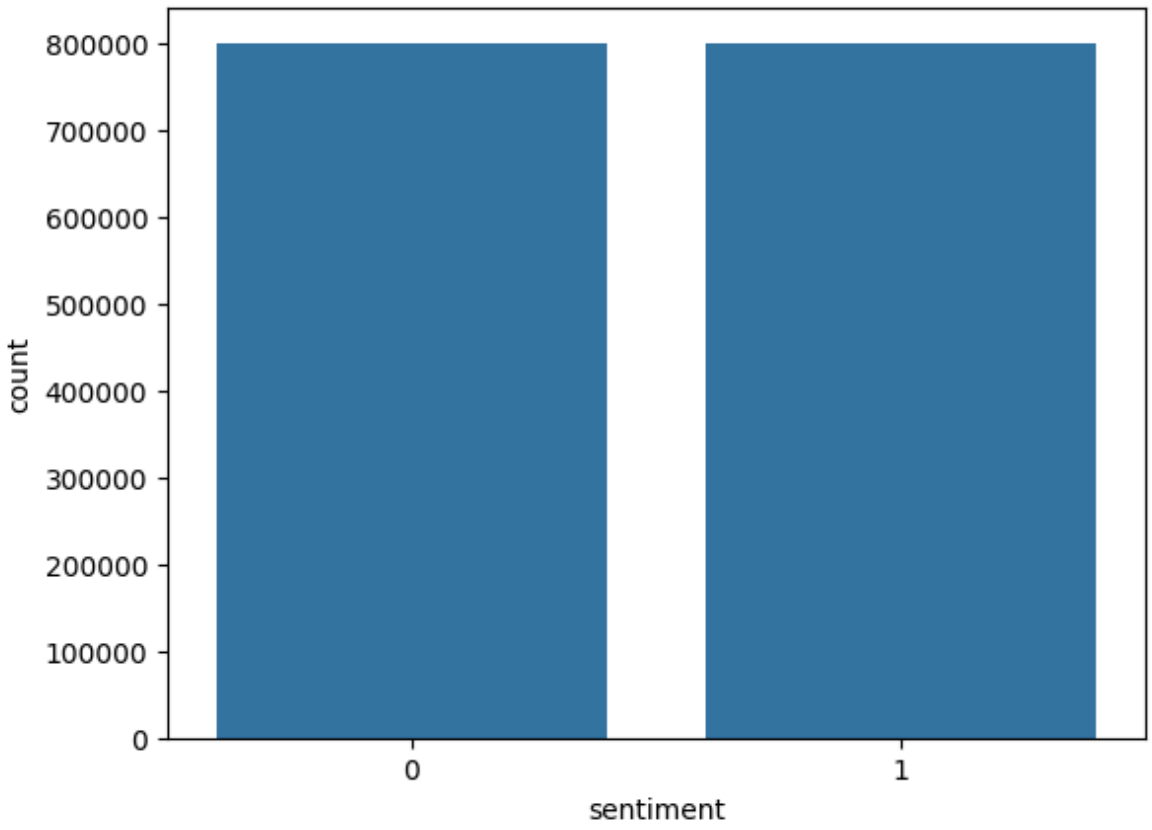
count values

In [14]:
```
data.sentiment.value_counts()
```

Out[14]:
```
sentiment
0    800000
1    800000
Name: count, dtype: int64
```

In [15]:
```
sns.countplot(x=data['sentiment'])
```

Out[15]:
```
<Axes: xlabel='sentiment', ylabel='count'>
```



Working on text columns

In [16]:
```
data.text.value_counts()
```

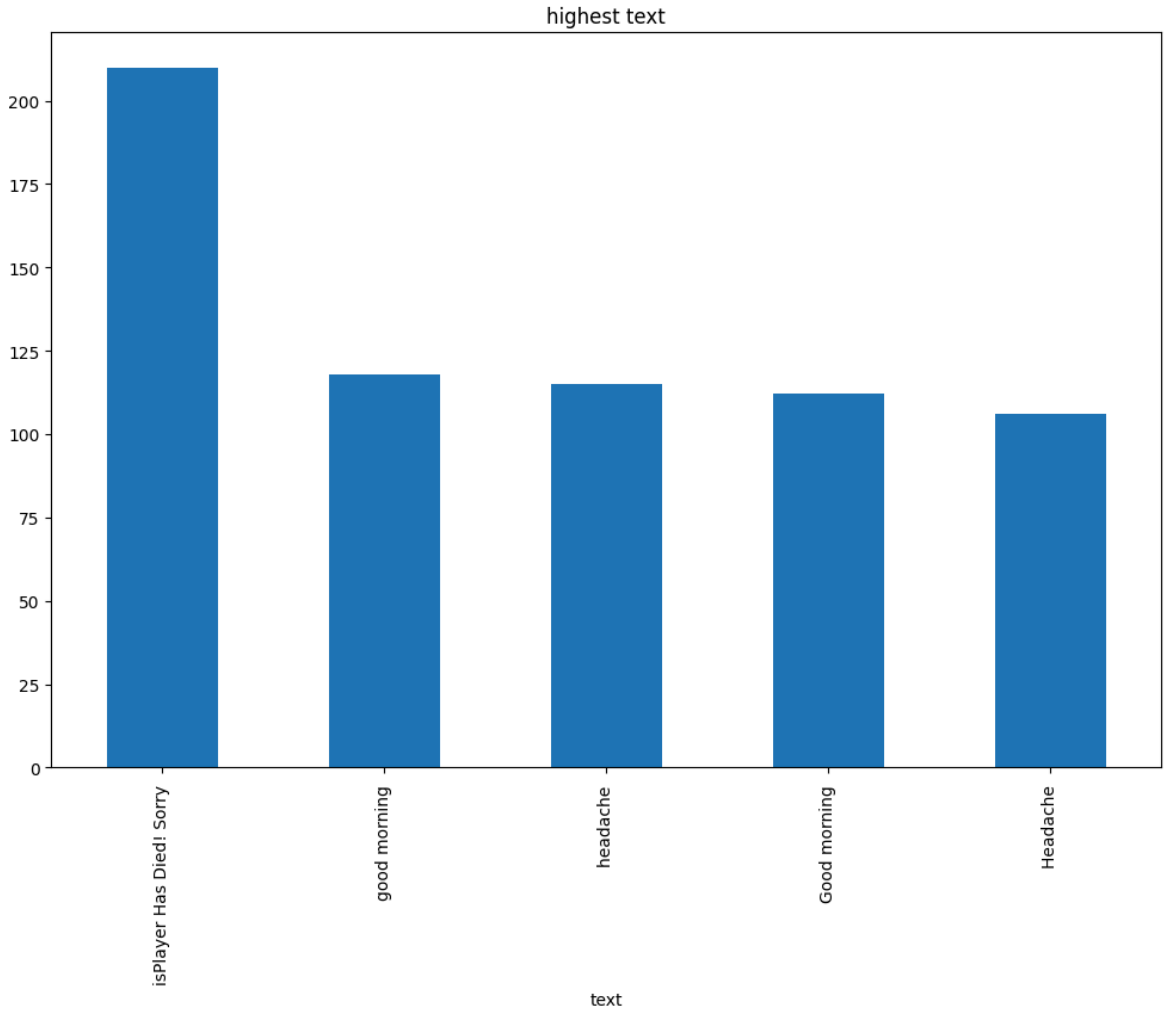
Out[16]:
```
text
isPlayer Has Died! Sorry
210
good morning
118
headache
115
Good morning
112
Headache
106

...
one of my friend called me, and asked to meet with her at Mid Valley today...bu
t i've no time *sigh*       1
@angry_barista I baked you a cake but I ated it
1
this week is not going as i had hoped
1
blagh class at 8 tomorrow
1
@LOLTrish hey  long time no see! Yes.. Rains a bit ,only a bit  LOL , I'm fine
thanks , how's you ?        1
Name: count, Length: 1581466, dtype: int64
```

Top no.of text

In [17]:
```
data.text.value_counts().head().plot.bar(figsize=(12,8))
plt.title("highest text")
```

Out[17]:
```
Text(0.5, 1.0, 'highest text')
```



Checking NAN values

In [18]:
```
data.isna().sum()
```

Out[18]:
```
sentiment    0
ids          0
date         0
flag         0
user         0
text         0
dtype: int64
```

date column covert object into int

In [19]:
```
data["date"]=pd.to_datetime(data["date"])
```

In [20]:
```
data.head()
```

Out[20]:

	sentiment	ids	date	flag	user	text
0	0	1467810369	2009-04-06 22:19:45	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	2009-04-06 22:19:49	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	2009-04-06 22:19:53	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	2009-04-06 22:19:57	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	2009-04-06 22:19:57	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

Access year

In [21]:
```
year = data.date.dt.year
```

In [22]:
```
year.unique()
```

Out[22]:
```
array([2009], dtype=int32)
```

Access time

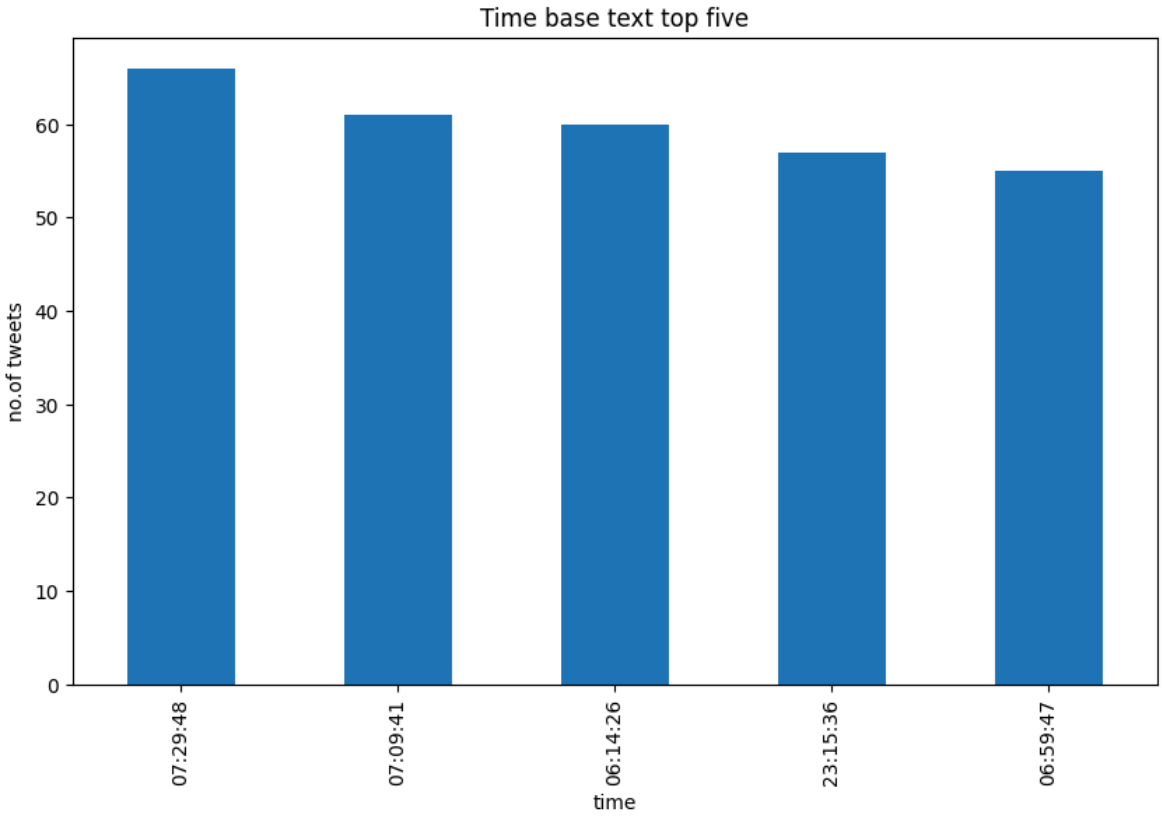
In [23]:
```
time = data.date.dt.time
```

In [24]:
```
time.value_counts()
```

Out[24]:
```
date
07:29:48    66
07:09:41    61
06:14:26    60
23:15:36    57
06:59:47    55
            ..
13:53:47     1
12:36:54     1
14:11:50     1
12:12:36     1
17:30:34     1
Name: count, Length: 86386, dtype: int64
```

In [25]:
```
time.value_counts().head().plot.bar(figsize=(10,6))
plt.title("Time base text top five")
plt.xlabel("time")
plt.ylabel("no.of tweets")
```

Out[25]: Text(0, 0.5, 'no.of tweets')



In [26]:
```
data = data[["sentiment", "text"]]
```

Stop removal wods

In [27]:
```
stop_words = set(stopwords.words('english'))
```

In [28]:
```
def preprocess_text(text):
    text = re.sub(r'\W', ' ', text)
    text = re.sub(r'\d', ' ', text)
    text = text.lower()
    text = ' '.join([word for word in text.split() if word not in stop_words])
    return text
```

In [29]:
```
data['text'] = data['text'].apply(preprocess_text)
```

In [30]:
```
data.text
```

Out[30]:
```
0          switchfoot http twitpic com zl awww bummer sho...
1          upset update facebook texting might cry result...
2          kenichan dived many times ball managed save re...
3                           whole body feels itchy like fire
4                           nationwideclass behaving mad see
                               ...
1599995                       woke school best feeling ever
1599996    thewdb com cool hear old walt interviews â htt...
1599997                     ready mojo makeover ask details
1599998    happy th birthday boo alll time tupac amaru sh...
1599999    happy charitytuesday thenspcc sparkscharity sp...
Name: text, Length: 1600000, dtype: object
```

getting Negative text

In [31]:
```
negative_texts = data[data['sentiment'] == 0]['text']
print("===== Negative Texts =====")
for i, text in enumerate(negative_texts.head(10)):  # Display first 10 negative
    print(f"{i+1}. {text}")
```

```
===== Negative Texts =====
1. switchfoot http twitpic com zl awww bummer shoulda got david carr third day
2. upset update facebook texting might cry result school today also blah
3. kenichan dived many times ball managed save rest go bounds
4. whole body feels itchy like fire
5. nationwideclass behaving mad see
6. kwesidei whole crew
7. need hug
8. loltrish hey long time see yes rains bit bit lol fine thanks
9. tatiana_k nope
10. twittera que muera
```

Getting positive text

In [32]:
```
positive_texts = data[data['sentiment'] == 1]['text']
print("\n===== Positive Texts =====")
for i, text in enumerate(positive_texts.head(10)):  # Display first 10 positive
    print(f"{i+1}. {text}")
```

```
===== Positive Texts =====
1. love health uandpets u guys r best
2. im meeting one besties tonight cant wait girl talk
3. darealsunisakim thanks twitter add sunisa got meet hin show dc area sweetheart
4. sick really cheap hurts much eat real food plus friends make soup
5. lovesbrooklyn effect everyone
6. productoffear tell burst laughing really loud thanks making come sulk
7. r_keith_hill thans response ihad already find answer
8. keepinupwkris jealous hope great time vegas like acm love show
9. tommcfly ah congrats mr fletcher finally joining twitter
10. e voip responded stupid cat helping type forgive errors
```

Apply Nltk and DL technique

In [33]:
```
tokenizer = Tokenizer(num_words=5000, oov_token="<OOV>")
```

In [34]:
```
tokenizer.fit_on_texts(data['text'])
```

In [35]:
```
tokenizer
```

Out[35]:
```
<keras.src.legacy.preprocessing.text.Tokenizer at 0x1f50b246de0>
```

In [36]:
```
sequences = tokenizer.texts_to_sequences(data['text'])
```

In [37]:
```
padded_sequences = pad_sequences(sequences, maxlen=50, padding='post')
```

In [38]:
```
X_train, X_test, y_train, y_test = train_test_split(padded_sequences,data['senti
```

Access train and test size

In [39]:
```
print(f"Training samples: {len(X_train)}, Testing samples: {len(X_test)}")
```

```
Training samples: 1280000, Testing samples: 320000
```

Creating a layers

In [40]:
```
def build_lstm_model():
    model = Sequential([
        Embedding(input_dim=5000, output_dim=64, input_length=50),
        LSTM(128, return_sequences=True),
        LSTM(64),
        Dense(64, activation='relu'),
        Dropout(0.5),
        Dense(1, activation='sigmoid')

    ])

    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accura
    return model
```

In [41]:
```
lstm_model = build_lstm_model()
```

summary

In [42]:
```
lstm_model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	
embedding (Embedding)	?	
lstm (LSTM)	?	
lstm_1 (LSTM)	?	
dense (Dense)	?	
dropout (Dropout)	?	
dense_1 (Dense)	?	

Total params: 0 (0.00 B)

Trainable params: 0 (0.00 B)

Non-trainable params: 0 (0.00 B)

In [43]:
```
lstm_model.fit(X_train, y_train, epochs=3, validation_data=(X_test, y_test), bat
```

```
Epoch 1/3
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 2568s 128ms/step - accuracy: 0.5002 - loss: 0.69
33 - val_accuracy: 0.5016 - val_loss: 0.6931
Epoch 2/3
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 10449s 522ms/step - accuracy: 0.5152 - loss: 0.6
825 - val_accuracy: 0.7749 - val_loss: 0.4670
Epoch 3/3
20000/20000 ━━━━━━━━━━━━━━━━━━━━ 2457s 123ms/step - accuracy: 0.7784 - loss: 0.46
75 - val_accuracy: 0.7840 - val_loss: 0.4523
```

Out[43]:
```
<keras.src.callbacks.history.History at 0x1f552934830>
```

save the model

In [68]:
```
lstm_model.save("my_model.keras")
```

TF_IDF

In [45]:
```
vectorizer = TfidfVectorizer()
```

In [46]:
```
vectorizer
```

Out[46]:
```
TfidfVectorizer()
```

In [47]:
```
X_train_tfidf = vectorizer.fit_transform(data["text"])
```

In [48]:
```
X_train_tfidf
```

Out[48]:
```
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 11828773 stored elements and shape (1600000, 615285)>
```

In [49]:
```
X_test_tfidf = vectorizer.transform(data["text"])
```

In [50]:
```
X_test_tfidf
```

Out[50]:
```
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 11828773 stored elements and shape (1600000, 615285)>
```

Model Creation

In [52]:
```
model = MultinomialNB()
```

In [53]:
```
model.fit(X_train_tfidf, data["sentiment"])
```

Out[53]:
```
MultinomialNB()
```

prediction

In [54]:
```
y_pred = model.predict(X_test_tfidf)
```

In [55]:
```
y_pred
```

Out[55]:
```
array([0, 0, 0, ..., 1, 1, 1])
```

accuracy score

In [56]:
```
accuracy = accuracy_score(data["sentiment"], y_pred)
```

In [57]:
```
print("Naïve Bayes Accuracy:", accuracy)
```

```
Naïve Bayes Accuracy: 0.827858125
```

In [58]:
```
print(classification_report(data["sentiment"], y_pred, target_names=['Negative',
```

```
              precision    recall  f1-score   support

    Negative       0.81      0.85      0.83    800000
    Positive       0.84      0.81      0.82    800000

    accuracy                           0.83   1600000
   macro avg       0.83      0.83      0.83   1600000
weighted avg       0.83      0.83      0.83   1600000
```

In [67]:
```
new_text = ["I love this project!", "This is the worst experience ever."]
```

In [60]:
```
new_text_processed = [preprocess_text(text) for text in new_text]
```

In [61]:
```
new_text_tfidf = vectorizer.transform(new_text_processed)
```

In [62]:
```
predictions = model.predict(new_text_tfidf)
```

In [63]:
```
predictions
```

Out[63]:
```
array([1, 0])
```

In [64]:
```
for text, prediction in zip(new_text, predictions):
    print(f"Text: {text} -> Sentiment: {'Positive' if prediction == 1 else 'Nega
```

```
Text: I love this project! -> Sentiment: Positive
Text: This is the worst experience ever. -> Sentiment: Negative
```

Visualation of -ve nad +ve tweets

In [65]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Combine all negative texts into a single string
negative_text = " ".join(negative_texts)
positive_text = " ".join(positive_texts)

# Generate word clouds
negative_wordcloud = WordCloud(width=800, height=400, background_color='white').
positive_wordcloud = WordCloud(width=800, height=400, background_color='white').

# Plot the word clouds
plt.figure(figsize=(12, 6))

# Negative word cloud
plt.subplot(1, 2, 1)
plt.imshow(negative_wordcloud, interpolation='bilinear')
plt.title("Negative Texts Word Cloud")
plt.axis('off')

# Positive word cloud
plt.subplot(1, 2, 2)
plt.imshow(positive_wordcloud, interpolation='bilinear')
plt.title("Positive Texts Word Cloud")
plt.axis('off')

plt.show()
```