```
# Importing necessary libraries
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import pandas as pd
import re

# Download NLTK resources
nltk.download('punkt')
nltk.download('stopwords')

# Load your dataset (replace the file path with your actual file)
df = pd.read_csv("https://github.com/suhasmaddali/Twitter-Sentiment-Analysis/raw/refs/heads/main/train.csv")

# Ensure all non-string values are handled
df['selected_text'] = df['selected_text'].fillna('').astype(str)
```

Follow link (cmd + click)

Loading...

```
# Step 1: Cleaning the text
cleaned = []
for text in df['selected_text']:
    cleaned_text = re.sub(r'[^\w\s]', '', text)  # Remove punctuation and special characters
    cleaned_text = re.sub(r'\s+', ' ', cleaned_text)  # Replace multiple spaces with a single space
    cleaned.append(cleaned_text.strip())  # Strip leading/trailing spaces

# Step 2: Tokenizing the cleaned text
tokens = [word_tokenize(x) for x in cleaned]

# Step 3: Removing stopwords
stop = set(stopwords.words('english'))
stpktn = [[word for word in sentence if word not in stop] for sentence in tokens]

# Step 4: Displaying results
print("Cleaned Text:", cleaned[:5])  # Display first 5 cleaned texts
print("Tokens:", tokens[:5])  # Display tokens for the first 5 cleaned texts
print("Tokens without Stopwords:", stpktn[:5])  # Display tokens without stopwords for the first 5 cleaned texts
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Cleaned Text: ['Id have responded if I were going', 'Sooo SAD', 'bullying me', 'leave me alone', 'Sons of']
Tokens: [['Id', 'have', 'responded', 'if', 'I', 'were', 'going'], ['Sooo', 'SAD'], ['bullying', 'me'], ['leave', 'me', 'alone'], ['Sons', 'of']]
Tokens without Stopwords: [['Id', 'responded', 'I', 'going'], ['Sooo', 'SAD'], ['bullying'], ['leave', 'alone'], ['Sons']]
```

```
# Import necessary libraries
import pandas as pd
import os
```

```
# Download the file from GitHub
!wget https://github.com/suhasmaddali/Twitter-Sentiment-Analysis/raw/refs/heads/main/train.csv -O train.csv

# Load the dataset into a Pandas DataFrame
df = pd.read_csv("train.csv")
```

```
--2025-01-24 13:34:32--  https://github.com/suhasmaddali/Twitter-Sentiment-Analysis/raw/refs/heads/main/train.csv
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/suhasmaddali/Twitter-Sentiment-Analysis/refs/heads/main/train.csv [following]
--2025-01-24 13:34:33--  https://raw.githubusercontent.com/suhasmaddali/Twitter-Sentiment-Analysis/refs/heads/main/train.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3501243 (3.3M) [text/plain]
Saving to: 'train.csv'

train.csv           100%[===================>]   3.34M  --.-KB/s    in 0.06s

2025-01-24 13:34:33 (54.6 MB/s) - 'train.csv' saved [3501243/3501243]
```

Follow link (cmd + click)

```
from sklearn.feature_extraction.text import CountVectorizer

cv=CountVectorizer()

#importing multinomialnb
from sklearn.naive_bayes import MultinomialNB

print(df.columns)
```

```
Index(['textID', 'text', 'selected_text', 'sentiment'], dtype='object')
```

```
y=df['selected_text']

y
```

	selected_text
0	I`d have responded, if I were going
1	Sooo SAD
2	bullying me
3	leave me alone
4	Sons of ****,
...	...
27476	d lost
27477	, don`t force
27478	Yay good for both of you.
27479	But it was worth it ****.
27480	All this flirting going on - The ATG smiles. Y...

27481 rows × 1 columns

dtype: object

Follow link (cmd + click)
Loading...

```
mb= MultinomialNB()
```

```
from nltk.stem import PorterStemmer
ps=PorterStemmer()
```

```
ps.stem(df['selected_text'][0])
```

```
'i`d have responded, if i were go'
```

```
 stemed_data=[]
 for message in stpktn:
  stem=[ps.stem(word) for word in message]
  stemed_data.append(stem)
```

```
stemed_data
```

```
          'today'],
         ['good'],
         ['awesom'],
         ['im', 'afraid'],
         ['im', 'best', 'day'],
         ['_aid16', 'goodnight'],
         ['welcom'],
         ['thank'],
         ['thank'],
         ['feel', 'home', 'freddi', 'funpack', 'day'],
         ['haha', 'rememb'],
         ['hee', 'exam', 'ill', 'give', 'u', 'plenti', 'show', 'haha', 'hopeless'],
         ['i', 'sorri'],
         ['awww', 'remind', 'get', 'readi', 'ball'],
         ['couldnt', 'rememb'],
         ['_bishop',
          'sez',
          'need',
          'get',
          'u',
          'look',
          'flight',
          'email',
          'ur',
          'work',
          'tonit'],
         ['i', 'wish', 'i', 'batteri', 'life', 'iphon'],
         ['happi'],
         ['omg'],
         ['better'],
         ['work', 'heat', 'horribl'],
         ['turn', 'human'],
         ['glad'],
         ['_starr',
          'yep',
          'hawkesburi',
          'classic',
          'start',
          'windsor',
          'home',
          'said',
          'dj'],
         ['thank', 'savvv'],
         ['sleep', 'pattern', 'screw'],
         ['isnt', 'easi', 'find'],
         ['may', 'fourth', 'happi', 'star', 'war', 'day'],
         ['lucki', 'never'],
         ...]
```

Follow link (cmd + click)

Loading...

```
stem_vec=[' '.join(message) for message in stemed_data]
```

```
stem_vec
```

```
 'youtub',
'call mum ask icecream',
'_ lazi day play xbox drink tea',
'disappoint',
'hope',
'my mom text told rodney chase firefli backyard awwwww im miss',
'happi',
'thank',
'my leg kill know good pain',
'fun',
'h wish i could go',
'readi go home',
'need push diet last level not good last week lost 1 lb better gain',
'excit',
'my back kill it wont keep drop lowhop i got someon pick back tho lol',
'',
'mad rain got',
'i grandpar place',
'hi chile it 9 deg c winter comingc wait cali sun',
'dont feel good',
'i born',
'congradt ur show even tho wasnt lol',
'damjust finish watch prison break final breakomg dont think ive cri hard showfin understand final',
'excus',
'glad friday two class lazi afternoon bad isnt nice warm today',
'good',
'awesom',
'im afraid',
'im best day',
'_aid16 goodnight',
'welcom',
'thank',
'thank',
'feel home freddi funpack day',
'haha rememb',
'hee exam ill give u plenti show haha hopeless',
'i sorri',
'awww remind get readi ball',
'couldnt rememb',
'_bishop sez need get u look flight email ur work tonit',
'i wish i batteri life iphon',
'happi',
'omg',
'better',
'work heat horribl',
'turn human',
'glad',
'_starr yep hawkesburi classic start windsor home said dj',
'thank savvv',
'sleep pattern screw',
'isnt easi find',
'may fourth happi star war day',
'lucki never',
...]
```

Follow link (cmd + click)

Loading...

```
x_vec=cv.fit_transform(stem_vec).toarray()
```

```
x_vec
```

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

```
len(x_vec[0])
```

```
15592
```

Follow link (cmd + click)

Loading...

```
y=df['sentiment']
```

```
y
```

	sentiment
0	neutral
1	negative
2	negative
3	negative
4	negative
...	...
27476	negative
27477	negative
27478	positive
27479	positive
27480	neutral

27481 rows × 1 columns

dtype: object

```
mb= MultinomialNB()
```

```
mb.fit(x_vec,y)
```

```
▾ MultinomialNB
MultinomialNB()
```

```
x_vec[0]
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

```
df['selected_text'][0]
```

```
'I`d have responded, if I were going'
```

```
mb.predict([x_vec[0]])
```

```
array(['neutral'], dtype='<U8')
```

Follow link (cmd + click)

Loading...

```
#1- do train test split
#2-create a logistic regression model
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_vec,y,test_size=0.2)
```

```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
y_pred=lr.predict(x_test[0].reshape(1,-1))
print(y_pred)
```

```
['positive']
```

```
lr.score(x_test,y_test)
```

```
0.8153538293614699
```

Start coding or generate with AI.