DSC550-T301

Chitramoy Mukherjee

Week-3

Date: 12/11/2023

In [21]:

```python
import warnings
warnings.filterwarnings('ignore')

# Required python basic libraries

import numpy as np
import pandas as pd
import textblob
from textblob import TextBlob
import string
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk import download
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
import nltk




from os.path import basename, exists


def download(url):
    filename = basename(url)
    if not exists(filename):
        from urllib.request import urlretrieve

        local, _ = urlretrieve(url, filename)
        print("Downloaded " + local)

### Reading the labeledTrainData.tsv file into DataFrame
df = pd.read_csv("C:\\Users\\14024\\OneDrive\\Desktop\\MS-DSC\\DSC-550\Week-3\\labelec

# Display the first few rows of the DataFrame to ensure it's loaded properly
print(df.head())
```

```
       id  sentiment                                             review
0  5814_8          1  With all this stuff going down at the moment w...
1  2381_9          1  \The Classic War of the Worlds\" by Timothy Hi...
2  7759_3          0  The film starts with a manager (Nicholas Bell)...
3  3630_4          0  It must be assumed that those who praised this...
4  9495_8          1  Superbly trashy and wondrously unpretentious 8...
```

In [22]:

```python
# How many of each positive and negative reviews are there?

# Count the number of positive and negative reviews
num_positive_reviews = df[df['sentiment'] == 1].shape[0]
num_negative_reviews = df[df['sentiment'] == 0].shape[0]
```

```

# Display the counts
print("Number of Positive Reviews:", num_positive_reviews)
print("Number of Negative Reviews:", num_negative_reviews)
```

```
Number of Positive Reviews: 12500
Number of Negative Reviews: 12500
```

In [23]:

```
# Use TextBlob to classify each movie review as positive or negative. Assume that a po
def classify_sentiment(review):
    analysis = TextBlob(review)
    return 'positive' if analysis.polarity >= 0 else 'negative'

# Apply the sentiment classification to the 'review' column
df['predicted_sentiment'] = df['review'].apply(classify_sentiment)

# Display the DataFrame with the predicted sentiment
print(df[['review', 'predicted_sentiment']].head())
```

```
                                              review predicted_sentiment
0  With all this stuff going down at the moment w...            positive
1  \The Classic War of the Worlds\" by Timothy Hi...            positive
2  The film starts with a manager (Nicholas Bell)...            negative
3  It must be assumed that those who praised this...            positive
4  Superbly trashy and wondrously unpretentious 8...            negative
```

In [5]:

```
from sklearn.metrics import accuracy_score
# Function to classify sentiment using TextBlob
def classify_sentiment(text):
    analysis = TextBlob(text)
    return 1 if analysis.sentiment.polarity >= 0 else 0

# Apply sentiment classification to each review
df['predicted_sentiment'] = df['review'].apply(classify_sentiment)
print(df[['review', 'sentiment', 'predicted_sentiment']].head())

# Calculate accuracy
accuracy = accuracy_score(df['sentiment'], df['predicted_sentiment'])
print(f'Accuracy: {accuracy * 100:.2f}%')

# Compare with random guessing accuracy
random_guessing_accuracy = max(df['sentiment'].mean(), 1 - df['sentiment'].mean())
print(f'Random Guessing Accuracy: {random_guessing_accuracy * 100:.2f}%')

# Compare with random guessing
if accuracy > random_guessing_accuracy:
    print("The sentiment analysis model is better than random guessing.")
else:
    print("The sentiment analysis model is not better than random guessing.")
```

```
                                              review  sentiment  \
0  With all this stuff going down at the moment w...          1
1  \The Classic War of the Worlds\" by Timothy Hi...          1
2  The film starts with a manager (Nicholas Bell)...          0
3  It must be assumed that those who praised this...          0
4  Superbly trashy and wondrously unpretentious 8...          1

   predicted_sentiment
0                    1
1                    1
2                    0
3                    1
4                    0
Accuracy: 68.52%
Random Guessing Accuracy: 50.00%
The sentiment analysis model is better than random guessing.
```

In [44]:

```
pip install flair
```

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

  Downloading flair-0.13.0-py3-none-any.whl (387 kB)
     --------------------------------------- 387.2/387.2 kB 4.0 MB/s eta 0:00:00
Collecting segtok>=1.5.11
  Downloading segtok-1.5.11-py3-none-any.whl (24 kB)
Collecting gensim>=4.2.0
  Downloading gensim-4.3.2-cp39-cp39-win_amd64.whl (24.0 MB)
     ---------------------------------------- 24.0/24.0 MB 10.7 MB/s eta 0:00:00
Collecting langdetect>=1.0.9
  Downloading langdetect-1.0.9.tar.gz (981 kB)
     --------------------------------------- 981.5/981.5 kB 4.4 MB/s eta 0:00:00
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Collecting wikipedia-api>=0.5.7
  Downloading Wikipedia_API-0.6.0-py3-none-any.whl (14 kB)
Collecting gdown>=4.4.0
  Downloading gdown-4.7.1-py3-none-any.whl (15 kB)
Collecting conllu>=4.0
  Downloading conllu-4.5.3-py2.py3-none-any.whl (16 kB)
Collecting ftfy>=6.1.0
  Downloading ftfy-6.1.3-py3-none-any.whl (53 kB)
     ----------------------------------------- 53.4/53.4 kB 2.7 MB/s eta 0:00:00
Requirement already satisfied: lxml>=4.8.0 in c:\users\14024\anaconda3\lib\site-packa
ges (from flair) (4.9.1)
Collecting deprecated>=1.2.13
  Downloading Deprecated-1.2.14-py2.py3-none-any.whl (9.6 kB)
Collecting transformer-smaller-training-vocab>=0.2.3
  Downloading transformer_smaller_training_vocab-0.3.3-py3-none-any.whl (14 kB)
Collecting mpld3>=0.3
  Downloading mpld3-0.5.9-py3-none-any.whl (201 kB)
     --------------------------------------- 201.2/201.2 kB 1.4 MB/s eta 0:00:00
Requirement already satisfied: tqdm>=4.63.0 in c:\users\14024\anaconda3\lib\site-pack
ages (from flair) (4.64.1)
Collecting semver<4.0.0,>=3.0.0
  Downloading semver-3.0.2-py3-none-any.whl (17 kB)
Collecting sqlitedict>=2.0.0
  Downloading sqlitedict-2.1.0.tar.gz (21 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\14024\anaconda3\lib
\site-packages (from flair) (2.8.2)
Collecting huggingface-hub>=0.10.0
  Downloading huggingface_hub-0.19.4-py3-none-any.whl (311 kB)
     --------------------------------------- 311.7/311.7 kB 9.7 MB/s eta 0:00:00
Collecting torch!=1.8,>=1.5.0
  Downloading torch-2.1.2-cp39-cp39-win_amd64.whl (192.2 MB)
     --------------------------------------- 192.2/192.2 MB 4.1 MB/s eta 0:00:00
Collecting bpemb>=0.3.2
  Downloading bpemb-0.3.4-py3-none-any.whl (19 kB)
Requirement already satisfied: tabulate>=0.8.10 in c:\users\14024\anaconda3\lib\site-
packages (from flair) (0.8.10)
Requirement already satisfied: boto3>=1.20.27 in c:\users\14024\anaconda3\lib\site-pa
ckages (from flair) (1.24.28)
Collecting more-itertools>=8.13.0
  Downloading more_itertools-10.1.0-py3-none-any.whl (55 kB)
     ----------------------------------------- 55.8/55.8 kB 3.0 MB/s eta 0:00:00
Collecting pptree>=3.1
  Downloading pptree-3.1.tar.gz (3.0 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: regex>=2022.1.18 in c:\users\14024\anaconda3\lib\site-
packages (from flair) (2022.7.9)
Collecting transformers[sentencepiece]<5.0.0,>=4.18.0
  Downloading transformers-4.36.1-py3-none-any.whl (8.3 MB)
     ---------------------------------------- 8.3/8.3 MB 3.1 MB/s eta 0:00:00
Collecting pytorch-revgrad>=0.2.0
  Downloading pytorch_revgrad-0.2.0-py3-none-any.whl (4.6 kB)
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\14024\anaconda3\lib\si
te-packages (from flair) (1.0.2)
Requirement already satisfied: urllib3<2.0.0,>=1.0.0 in c:\users\14024\anaconda3\lib
\site-packages (from flair) (1.26.11)
Requirement already satisfied: matplotlib>=2.2.3 in c:\users\14024\anaconda3\lib\site
-packages (from flair) (3.5.2)
Collecting janome>=0.4.2
  Downloading Janome-0.5.0-py2.py3-none-any.whl (19.7 MB)
     ---------------------------------------- 19.7/19.7 MB 4.5 MB/s eta 0:00:00
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in c:\users\14024\anaconda3\lib
\site-packages (from boto3>=1.20.27->flair) (0.10.0)
Requirement already satisfied: s3transfer<0.7.0,>=0.6.0 in c:\users\14024\anaconda3\l
ib\site-packages (from boto3>=1.20.27->flair) (0.6.0)
Requirement already satisfied: botocore<1.28.0,>=1.27.28 in c:\users\14024\anaconda3
\lib\site-packages (from boto3>=1.20.27->flair) (1.27.28)
Requirement already satisfied: numpy in c:\users\14024\anaconda3\lib\site-packages (f
rom bpemb>=0.3.2->flair) (1.21.5)
Collecting sentencepiece
  Downloading sentencepiece-0.1.99-cp39-cp39-win_amd64.whl (977 kB)
     -------------------------------------- 977.6/977.6 kB 3.9 MB/s eta 0:00:00
Requirement already satisfied: requests in c:\users\14024\anaconda3\lib\site-packages
(from bpemb>=0.3.2->flair) (2.28.1)
Requirement already satisfied: wrapt<2,>=1.10 in c:\users\14024\anaconda3\lib\site-pa
ckages (from deprecated>=1.2.13->flair) (1.14.1)
Collecting wcwidth<0.3.0,>=0.2.12
  Downloading wcwidth-0.2.12-py2.py3-none-any.whl (34 kB)
Requirement already satisfied: beautifulsoup4 in c:\users\14024\anaconda3\lib\site-pa
ckages (from gdown>=4.4.0->flair) (4.11.1)
Requirement already satisfied: filelock in c:\users\14024\anaconda3\lib\site-packages
(from gdown>=4.4.0->flair) (3.6.0)
Requirement already satisfied: six in c:\users\14024\anaconda3\lib\site-packages (fro
m gdown>=4.4.0->flair) (1.16.0)
Requirement already satisfied: scipy>=1.7.0 in c:\users\14024\anaconda3\lib\site-pack
ages (from gensim>=4.2.0->flair) (1.9.1)
Requirement already satisfied: smart-open>=1.8.1 in c:\users\14024\anaconda3\lib\site
-packages (from gensim>=4.2.0->flair) (5.2.1)
Requirement already satisfied: packaging>=20.9 in c:\users\14024\anaconda3\lib\site-p
ackages (from huggingface-hub>=0.10.0->flair) (21.3)
Collecting fsspec>=2023.5.0
  Downloading fsspec-2023.12.2-py3-none-any.whl (168 kB)
     -------------------------------------- 169.0/169.0 kB 9.9 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\14024\anaconda3
\lib\site-packages (from huggingface-hub>=0.10.0->flair) (4.3.0)
Requirement already satisfied: pyyaml>=5.1 in c:\users\14024\anaconda3\lib\site-packa
ges (from huggingface-hub>=0.10.0->flair) (6.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\14024\anaconda3\lib\site-pac
kages (from matplotlib>=2.2.3->flair) (9.2.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\14024\anaconda3\lib\site-
packages (from matplotlib>=2.2.3->flair) (3.0.9)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\14024\anaconda3\lib\site
-packages (from matplotlib>=2.2.3->flair) (4.25.0)
Requirement already satisfied: cycler>=0.10 in c:\users\14024\anaconda3\lib\site-pack
```

```
ages (from matplotlib>=2.2.3->flair) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\14024\anaconda3\lib\site
-packages (from matplotlib>=2.2.3->flair) (1.4.2)
Requirement already satisfied: jinja2 in c:\users\14024\anaconda3\lib\site-packages
(from mpld3>=0.3->flair) (2.11.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\14024\anaconda3\lib\s
ite-packages (from scikit-learn>=1.0.2->flair) (2.2.0)
Requirement already satisfied: joblib>=0.11 in c:\users\14024\anaconda3\lib\site-pack
ages (from scikit-learn>=1.0.2->flair) (1.1.0)
Requirement already satisfied: sympy in c:\users\14024\anaconda3\lib\site-packages (f
rom torch!=1.8,>=1.5.0->flair) (1.10.1)
Requirement already satisfied: networkx in c:\users\14024\anaconda3\lib\site-packages
(from torch!=1.8,>=1.5.0->flair) (2.8.4)
Requirement already satisfied: colorama in c:\users\14024\anaconda3\lib\site-packages
(from tqdm>=4.63.0->flair) (0.4.5)
Collecting tokenizers<0.19,>=0.14
  Downloading tokenizers-0.15.0-cp39-none-win_amd64.whl (2.2 MB)
     ---------------------------------------- 2.2/2.2 MB 3.2 MB/s eta 0:00:00
Collecting safetensors>=0.3.1
  Downloading safetensors-0.4.1-cp39-none-win_amd64.whl (277 kB)
     -------------------------------------- 277.8/277.8 kB 4.3 MB/s eta 0:00:00
Collecting protobuf
  Downloading protobuf-4.25.1-cp39-cp39-win_amd64.whl (413 kB)
     -------------------------------------- 413.4/413.4 kB 5.2 MB/s eta 0:00:00
Collecting accelerate>=0.21.0
  Downloading accelerate-0.25.0-py3-none-any.whl (265 kB)
     -------------------------------------- 265.7/265.7 kB 5.4 MB/s eta 0:00:00
Requirement already satisfied: soupsieve>1.2 in c:\users\14024\anaconda3\lib\site-pac
kages (from beautifulsoup4->gdown>=4.4.0->flair) (2.3.1)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\14024\anaconda3\lib\site-
packages (from jinja2->mpld3>=0.3->flair) (2.0.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\14024\anaconda3\lib\site-pack
ages (from requests->bpemb>=0.3.2->flair) (3.3)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\14024\anaconda3\l
ib\site-packages (from requests->bpemb>=0.3.2->flair) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\14024\anaconda3\lib\sit
e-packages (from requests->bpemb>=0.3.2->flair) (2022.9.14)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in c:\users\14024\anaconda3\lib
\site-packages (from requests->bpemb>=0.3.2->flair) (1.7.1)
Requirement already satisfied: mpmath>=0.19 in c:\users\14024\anaconda3\lib\site-pack
ages (from sympy->torch!=1.8,>=1.5.0->flair) (1.2.1)
Requirement already satisfied: psutil in c:\users\14024\anaconda3\lib\site-packages
(from accelerate>=0.21.0->transformers[sentencepiece]<5.0.0,>=4.18.0->flair) (5.9.0)
Building wheels for collected packages: langdetect, pptree, sqlitedict
  Building wheel for langdetect (setup.py): started
  Building wheel for langdetect (setup.py): finished with status 'done'
  Created wheel for langdetect: filename=langdetect-1.0.9-py3-none-any.whl size=99322
5 sha256=a3cf388b0018e2c805c2bff2f118cf0dfa21e61845ebc3aa5941a682da4c3eda
  Stored in directory: c:\users\14024\appdata\local\pip\cache\wheels\d1\c1\d9\7e068de
779d863bc8f8fc9467d85e25cfe47fa5051fff1a1bb
  Building wheel for pptree (setup.py): started
  Building wheel for pptree (setup.py): finished with status 'done'
  Created wheel for pptree: filename=pptree-3.1-py3-none-any.whl size=4609 sha256=e38
3d8cf4a836ff0d4f3af95a8db4e98a2f494a56052a6475385445f626720f0
  Stored in directory: c:\users\14024\appdata\local\pip\cache\wheels\52\0e\51\514e690
004ea9713bc3fdb678d5e2768fcc597d0c3b6a3abd2
  Building wheel for sqlitedict (setup.py): started
  Building wheel for sqlitedict (setup.py): finished with status 'done'
  Created wheel for sqlitedict: filename=sqlitedict-2.1.0-py3-none-any.whl size=16864
sha256=3ff2797b84a62ba7be9bc9982b1675bc06fb76f749b7d6aedd62b8ab06b8a17c
```

```
  Stored in directory: c:\users\14024\appdata\local\pip\cache\wheels\f6\48\c4\942f7a1
d556fddd2348cb9ac262f251873dfd8a39afec5678e
Successfully built langdetect pptree sqlitedict
Installing collected packages: wcwidth, sqlitedict, sentencepiece, pptree, janome, se
mver, segtok, safetensors, protobuf, more-itertools, langdetect, ftfy, fsspec, deprec
ated, conllu, wikipedia-api, torch, huggingface-hub, gensim, tokenizers, pytorch-revg
rad, mpld3, gdown, bpemb, accelerate, transformers, transformer-smaller-training-voca
b, flair
  Attempting uninstall: wcwidth
    Found existing installation: wcwidth 0.2.5
    Uninstalling wcwidth-0.2.5:
      Successfully uninstalled wcwidth-0.2.5
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2022.7.1
    Uninstalling fsspec-2022.7.1:
      Successfully uninstalled fsspec-2022.7.1
  Attempting uninstall: gensim
    Found existing installation: gensim 4.1.2
    Uninstalling gensim-4.1.2:
      Successfully uninstalled gensim-4.1.2
Successfully installed accelerate-0.25.0 bpemb-0.3.4 conllu-4.5.3 deprecated-1.2.14 f
lair-0.13.0 fsspec-2023.12.2 ftfy-6.1.3 gdown-4.7.1 gensim-4.3.2 huggingface-hub-0.1
9.4 janome-0.5.0 langdetect-1.0.9 more-itertools-10.1.0 mpld3-0.5.9 pptree-3.1 protob
uf-4.25.1 pytorch-revgrad-0.2.0 safetensors-0.4.1 segtok-1.5.11 semver-3.0.2 sentence
piece-0.1.99 sqlitedict-2.1.0 tokenizers-0.15.0 torch-2.1.2 transformer-smaller-train
ing-vocab-0.3.3 transformers-4.36.1 wcwidth-0.2.12 wikipedia-api-0.6.0
```

In [ ]:

```python
import pandas as pd
from flair.models import TextClassifier
from flair.data import Sentence
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

# Initialize the Flair text classifier (pre-trained model for sentiment analysis)
classifier = TextClassifier.load('en-sentiment')

# Classify each review using Flair
def classify_with_flair(review):
    sentence = Sentence(review)
    classifier.predict(sentence)
    return sentence.labels[0].value.lower()

df['Flair_Prediction'] = df['review'].apply(classify_with_flair)

# Map positive and negative labels to 1 and 0 for comparison
df['Flair_Prediction'] = df['Flair_Prediction'].map({'positive': 1, 'negative': 0})

# Check the accuracy of the Flair model
accuracy = accuracy_score(df['sentiment'], df['Flair_Prediction'])
print("\nAccuracy of the Flair model: {:.2%}".format(accuracy))

# Compare with random guessing
random_accuracy = 0.5  # Assuming a binary classification
print("Accuracy of random guessing: {:.2%}".format(random_accuracy))
```

```
2023-12-17 23:16:18,521 https://nlp.informatik.hu-berlin.de/resources/models/sentimen
t-curated-distilbert/sentiment-en-mix-distillbert_4.pt not found in cache, downloadin
g to C:\Users\14024\AppData\Local\Temp\tmp7o5grgc6
100%|████████████████████████████████████████████████████████████████████████████████|
253M/253M [00:46<00:00, 5.73MB/s]
```

```
2023-12-17 23:17:05,606 copying C:\Users\14024\AppData\Local\Temp\tmp7o5grgc6 to cach
e at C:\Users\14024\.flair\models\sentiment-en-mix-distillbert_4.pt
```

```
2023-12-17 23:17:05,972 removing temp file C:\Users\14024\AppData\Local\Temp\tmp7o5gr
gc6
tokenizer_config.json:   0%|          | 0.00/28.0 [00:00<?, ?B/s]
config.json:   0%|          | 0.00/483 [00:00<?, ?B/s]
vocab.txt:   0%|          | 0.00/232k [00:00<?, ?B/s]
tokenizer.json:   0%|          | 0.00/466k [00:00<?, ?B/s]
```

In [19]:

```
# Convert all text to lowercase
df['Review'] = df['review'].str.lower()

# Display the first few rows after converting to lowercase
print("\nFirst few rows of the DataFrame after converting to lowercase:")
print(df.head())
```

```
First few rows of the DataFrame after converting to lowercase:
       id  sentiment                                             review  \
0  5814_8          1  With all this stuff going down at the moment w...
1  2381_9          1  \The Classic War of the Worlds\" by Timothy Hi...
2  7759_3          0  The film starts with a manager (Nicholas Bell)...
3  3630_4          0  It must be assumed that those who praised this...
4  9495_8          1  Superbly trashy and wondrously unpretentious 8...

  predicted_sentiment                                             Review
0            positive  with all this stuff going down at the moment w...
1            positive  \the classic war of the worlds\" by timothy hi...
2            negative  the film starts with a manager (nicholas bell)...
3            positive  it must be assumed that those who praised this...
4            negative  superbly trashy and wondrously unpretentious 8...
```

In [25]:

```
# Function to remove punctuation and special characters
def remove_punctuation(text):
    # Use string.punctuation to get the set of all punctuation characters
    translator = str.maketrans('', '', string.punctuation)
    # Remove punctuation using the translator
    text_no_punct = text.translate(translator)
    return text_no_punct

# Remove punctuation and special characters from the 'Review' column
df['Cleaned_review'] = df['review'].apply(remove_punctuation)

# Display the DataFrame with the cleaned reviews
print("\nDataFrame with cleaned reviews:")
print(df[['review', 'Cleaned_review']].head())
```

```
DataFrame with cleaned reviews:
                                              review  \
0  With all this stuff going down at the moment w...
1  \The Classic War of the Worlds\" by Timothy Hi...
2  The film starts with a manager (Nicholas Bell)...
3  It must be assumed that those who praised this...
4  Superbly trashy and wondrously unpretentious 8...

                                      Cleaned_review
0  With all this stuff going down at the moment w...
1  The Classic War of the Worlds by Timothy Hines...
2  The film starts with a manager Nicholas Bell g...
3  It must be assumed that those who praised this...
4  Superbly trashy and wondrously unpretentious 8...
```

In [36]:

```python
# Remove stop words
stop_words = set(stopwords.words('english'))

def remove_stop_words(text):
    words = word_tokenize(text)
    filtered_words = [word.lower() for word in words if word.isalnum() and word.lower(
    return ' '.join(filtered_words)

df['Cleaned_review'] = df['review'].apply(remove_stop_words)

# Display the first few rows of the DataFrame with cleaned reviews
print("\nFirst few rows of the DataFrame with cleaned reviews:")
print(df[['review', 'Cleaned_review']].head())
```

```
First few rows of the DataFrame with cleaned reviews:
                                              review  \
0  With all this stuff going down at the moment w...
1  \The Classic War of the Worlds\" by Timothy Hi...
2  The film starts with a manager (Nicholas Bell)...
3  It must be assumed that those who praised this...
4  Superbly trashy and wondrously unpretentious 8...

                                      Cleaned_review
0  stuff going moment mj started listening music ...
1  classic war timothy hines entertaining film ob...
2  film starts manager nicholas bell giving welco...
3  must assumed praised film greatest filmed oper...
4  superbly trashy wondrously unpretentious 80 ex...
```

In [30]:

```python
# Initialize PorterStemmer
porter = PorterStemmer()

# Apply PorterStemmer to each review
df['Stemmed_review'] = df['review'].apply(lambda x: ' '.join([porter.stem(word) for wo

# Display the DataFrame with the new 'Stemmed_Review' column
print("\nDataFrame with Stemmed Reviews:")
print(df[['review', 'Stemmed_review']])

# Save the DataFrame with the stemmed reviews if needed
# df.to_csv('path_to_save_stemmed_reviews.csv', index=False)
```

```
DataFrame with Stemmed Reviews:
                                                  review  \
0      With all this stuff going down at the moment w...
1      \The Classic War of the Worlds\" by Timothy Hi...
2      The film starts with a manager (Nicholas Bell)...
3      It must be assumed that those who praised this...
4      Superbly trashy and wondrously unpretentious 8...
...                                                  ...
24995  It seems like more consideration has gone into...
24996  I don't believe they made this film. Completel...
24997  Guy is a loser. Can't get girls, needs to buil...
24998  This 30 minute documentary Buñuel made in the ...
24999  I saw this movie as a child and it broke my he...

                                          Stemmed_review
0      with all thi stuff go down at the moment with ...
1      \the classic war of the worlds\ '' by timothi ...
2      the film start with a manag ( nichola bell ) g...
3      it must be assum that those who prais thi film...
4      superbl trashi and wondrous unpretenti 80 's e...
...                                                  ...
24995  it seem like more consider ha gone into the im...
24996  i do n't believ they made thi film . complet u...
24997  guy is a loser . ca n't get girl , need to bui...
24998  thi 30 minut documentari buñuel made in the ea...
24999  i saw thi movi as a child and it broke my hear...

[25000 rows x 2 columns]
```

In [34]:

```
# Create a bag-of-words matrix from your stemmed text (output from (4)), where each r
# Initialize NLTK's PorterStemmer
porter_stemmer = PorterStemmer()

# Tokenize and apply stemming to each review
df['Stemmed_Review'] = df['review'].apply(lambda x: ' '.join([porter_stemmer.stem(wor

# Create a bag-of-words matrix using CountVectorizer
vectorizer = CountVectorizer()
bag_of_words_matrix = vectorizer.fit_transform(df['Stemmed_review'])

# Display the dimensions of the bag-of-words matrix
print("\nDimensions of the bag-of-words matrix:")
print("Number of Rows (Reviews):", bag_of_words_matrix.shape[0])
print("Number of Columns (Unique Words):", bag_of_words_matrix.shape[1])
```

```
Dimensions of the bag-of-words matrix:
Number of Rows (Reviews): 25000
Number of Columns (Unique Words): 59685
```

In [35]:

```
# Create a term frequency-inverse document frequency (tf-idf) matrix from your stemme
# Apply NLTK's PorterStemmer
porter = PorterStemmer()
df['Stemmed_review'] = df['review'].apply(lambda x: ' '.join([porter.stem(word) for w

# Create a bag-of-words matrix
vectorizer = CountVectorizer()
bow_matrix = vectorizer.fit_transform(df['Stemmed_review'])

# Display dimensions of the bag-of-words matrix
print("\nDimensions of the bag-of-words matrix:")
print("Rows (documents):", bow_matrix.shape[0])
print("Columns (unique words):", bow_matrix.shape[1])

# Create a tf-idf matrix
tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(df['Stemmed_review'])

# Display dimensions of the tf-idf matrix
print("\nDimensions of the tf-idf matrix:")
print("Rows (documents):", tfidf_matrix.shape[0])
print("Columns (unique words):", tfidf_matrix.shape[1])
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\14024\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
Dimensions of the bag-of-words matrix:
Rows (documents): 25000
Columns (unique words): 59685

Dimensions of the tf-idf matrix:
Rows (documents): 25000
Columns (unique words): 59685
```