**Step 1: Install Required Libraries**

```python

pip install nltk spacy pandas matplotlib scikit-learn

```

This line installs necessary libraries using `pip`.

**Step 2: Import Required Libraries**

```python

import nltk

import pandas as pd

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

from nltk.sentiment.vader import SentimentIntensityAnalyzer

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.decomposition import LatentDirichletAllocation

```

Imports the required libraries: NLTK, pandas, scikit-learn, and SpaCy for text processing, sentiment analysis, and topic modeling.

**Step 3: NLTK Configuration**

```python

nltk.download()

```

Initiates the NLTK downloader GUI to download necessary NLTK corpora or models interactively.

**Step 4: Load Dataset**

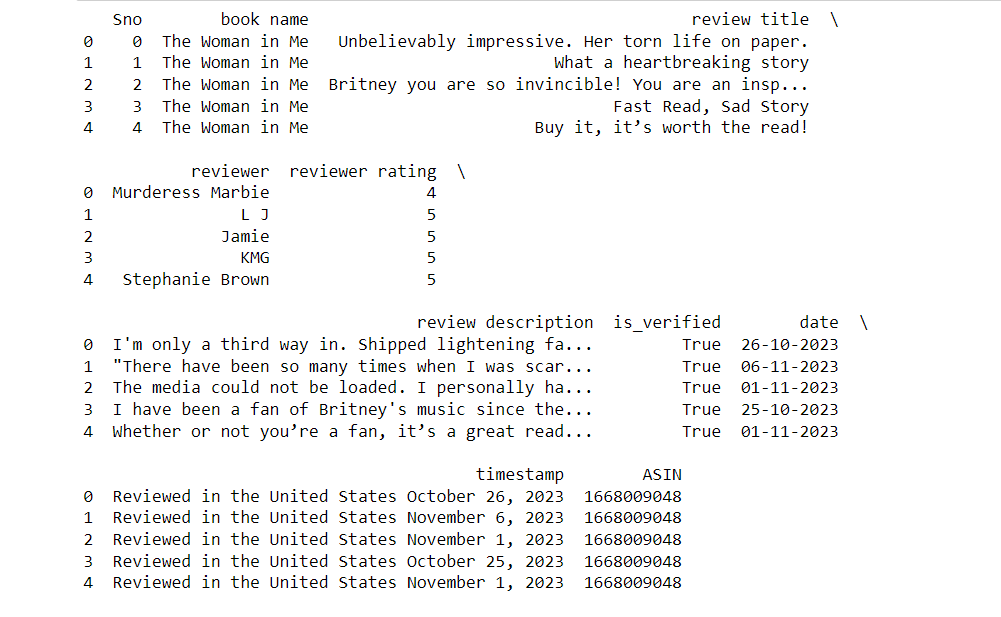
```python

data = pd.read\_csv(r'C:\Users\aishwarya\OneDrive\Desktop\Data Science\ML\Virtual\_Vigilantes\customer reviews.csv')

# Display the first few rows to understand the structure

print(data.head())

```



Loads the dataset into a Pandas DataFrame for further processing. Change the file path to the location of your CSV file.

**Step 5: Text Preprocessing (NLTK)**

```python

def preprocess\_text(text):

# Preprocessing steps: lowercase, tokenization, removing stopwords

# Apply this function to the 'review title' column

data['cleaned\_text'] = data['review title'].apply(preprocess\_text)

```

Preprocesses the 'review title' column using NLTK for tokenization and stopword removal.

**Step 6: SpaCy Configuration and Text Preprocessing**

```python

!python -m spacy download en\_core\_web\_sm

import spacy

# Load the 'en\_core\_web\_sm' model

nlp = spacy.load('en\_core\_web\_sm')

def preprocess\_text\_spacy(text):

# Preprocessing using SpaCy: tokenization, removing stop words and punctuation

# Apply this function to the 'review title' column

data['cleaned\_text\_spacy'] = data['review title'].apply(preprocess\_text\_spacy)

```

Downloads and loads the SpaCy model 'en\_core\_web\_sm'. Then, applies SpaCy-based preprocessing to the 'review title' column.

**Step 7: Sentiment Analysis (VADER)**

```python

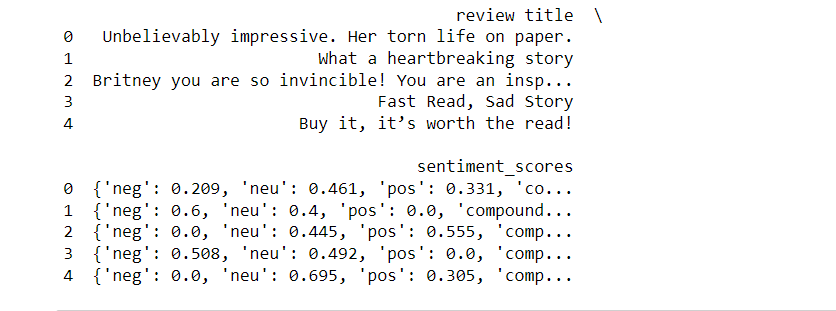
sid = SentimentIntensityAnalyzer()

# Calculate sentiment scores using VADER

data['sentiment\_scores'] = data['review title'].apply(lambda x: sid.polarity\_scores(x))

```

Initializes the VADER sentiment analyzer and calculates sentiment scores for each review title.



**Step 8: Topic Modeling (LDA)**

```python

vectorizer = CountVectorizer()

X = vectorizer.fit\_transform(data['review title'])

lda = LatentDirichletAllocation(n\_components=5, random\_state=42)

lda.fit(X)

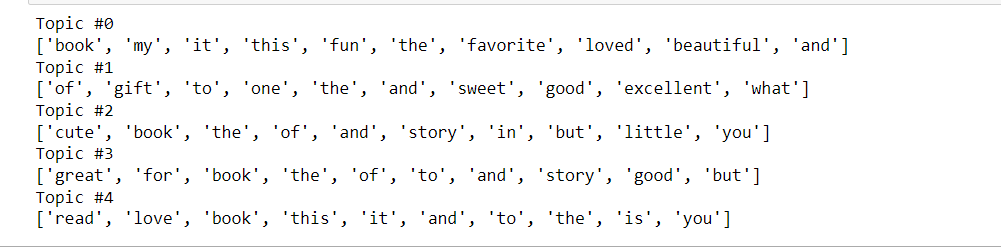
def display\_topics(model, feature\_names, n\_top\_words):

# Displays top words for each topic

# Uses 'get\_feature\_names\_out()' instead of 'get\_feature\_names()'

display\_topics(lda, feature\_names, n\_top\_words)

```



Applies Latent Dirichlet Allocation (LDA) for topic modeling on the 'review title' column and displays the top words for each topic.

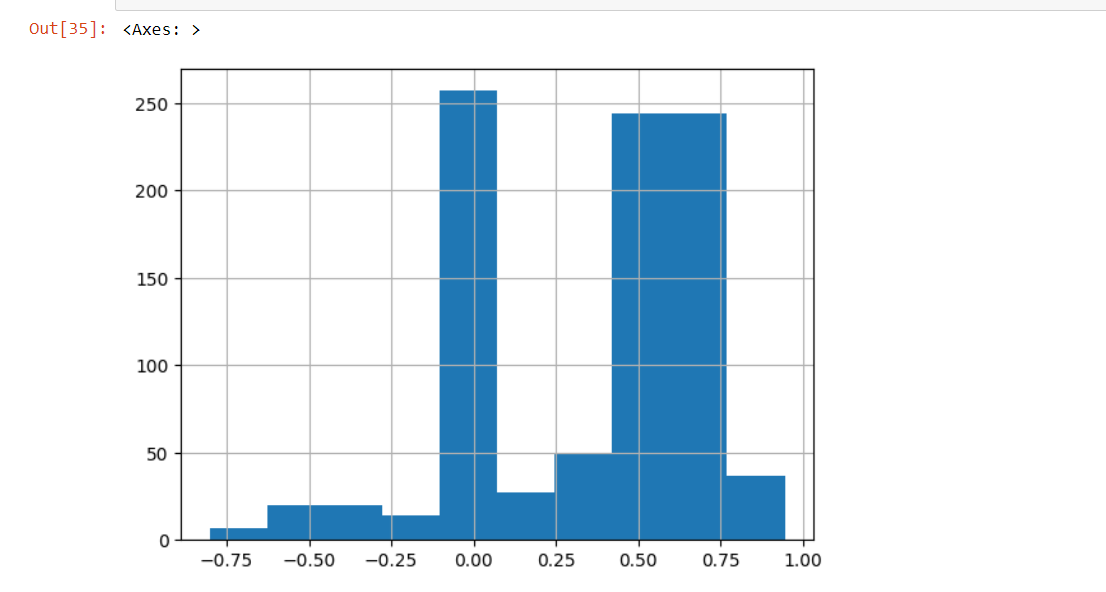
**Step 9: Visualization**

```python

data['sentiment\_scores'].apply(lambda x: x['compound']).hist()

```

Produces a histogram to visualize the sentiment distribution of the reviews.



This code performs various NLP tasks such as text preprocessing, sentiment analysis, and topic modeling using different libraries like NLTK, SpaCy, and scikit-learn.