```
!pip uninstall -y nltk
!pip install nltk
```

```
Found existing installation: nltk 3.9.1
Uninstalling nltk-3.9.1:
  Successfully uninstalled nltk-3.9.1
Collecting nltk
  Using cached nltk-3.9.1-py3-none-any.whl.metadata (2.9 kB)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
Using cached nltk-3.9.1-py3-none-any.whl (1.5 MB)
Installing collected packages: nltk
Successfully installed nltk-3.9.1
```

```
import pandas as pd
import os
import re
import nltk
from nltk.tokenize import word_tokenize
```

```
os.listdir('/content')
```

```
['.config', 'test.csv', 'sample_data']
```

```
import pandas as pd

df = pd.read_csv('/content/test.csv')
print(df.head())
```

```
                                                text
0  This movie was horrible. If it had never been ...
1  The director infuses this film with false dept...
2  I don't get it! The teenage leads in "Horror S...
3  This is the fifth part of 'The Animatrix', a c...
4  I was very impressed with with this film which...
```

Load the CSV file

```
df = pd.read_csv('/content/test.csv')
df
```

	text
0	This movie was horrible. If it had never been ...
1	The director infuses this film with false dept...
2	I don't get it! The teenage leads in "Horror S...
3	This is the fifth part of 'The Animatrix', a c...
4	I was very impressed with with this film which...
...	...
10996	In the wake of my personal research into the p...
10997	I had a bit of hope for this hour long film ma...
10998	Having been pleasantly surprised by Sandra Bul...
10999	Elfriede Jelinek, not quite a household name y...
11000	There's something rotten about this film, and ...

11001 rows × 1 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

Check column names

```
print("Columns in the dataset:")
print(df.columns)
```

```
Columns in the dataset:
Index(['text'], dtype='object')
```

basic information about the dataset

```
print("\nDataset Info:")
print(df.info())
```

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11001 entries, 0 to 11000
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   text    11000 non-null  object
dtypes: object(1)
memory usage: 86.1+ KB
None
```

numerical data

```
print("\nStatistical Summary of Numerical Data:")
print(df.describe())
```

```
Statistical Summary of Numerical Data:
                                                      text
count                                                11000
unique                                               10937
top     I see that C. Thomas Howell has appeared in ma...
freq                                                     3
```

```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
```

```
import nltk
nltk.download('punkt')
nltk.download('wordnet')
import nltk
nltk.download('punkt_tab')  # Just in case lemmatization is needed
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
True
```

```
import nltk
nltk.data.path.append('/usr/local/share/nltk_data')
```

```
import re
from nltk.tokenize import word_tokenize

def preprocess_text(text):
    if not isinstance(text, str):
        return ""
    text = text.lower()
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    tokens = word_tokenize(text)
    return " ".join(tokens)

df['cleaned_review'] = df['text'].apply(preprocess_text)
print(df[['text', 'cleaned_review']].head())
```

```
                                                text  \
0  This movie was horrible. If it had never been ...
1  The director infuses this film with false dept...
2  I don't get it! The teenage leads in "Horror S...
3  This is the fifth part of 'The Animatrix', a c...
```

```
4  I was very impressed with with this film which...

                                        cleaned_review
0  this movie was horrible if it had never been m...
1  the director infuses this film with false dept...
2  i dont get it the teenage leads in horror star...
3  this is the fifth part of the animatrix a coll...
4  i was very impressed with with this film which...
```

```
df = df.dropna(subset=['text'])  # Remove rows where 'text' is NaN
df['cleaned_review'] = df['text'].apply(preprocess_text)
print(df[['text', 'cleaned_review']].head())
```

```
                                                text  \
0  This movie was horrible. If it had never been ...
1  The director infuses this film with false dept...
2  I don't get it! The teenage leads in "Horror S...
3  This is the fifth part of 'The Animatrix', a c...
4  I was very impressed with with this film which...

                                        cleaned_review
0  this movie was horrible if it had never been m...
1  the director infuses this film with false dept...
2  i dont get it the teenage leads in horror star...
3  this is the fifth part of the animatrix a coll...
4  i was very impressed with with this film which...
<ipython-input-15-430af35fd680>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df['cleaned_review'] = df['text'].apply(preprocess_text)
```

```
import re
import nltk
from nltk.tokenize import word_tokenize
```

```
nltk.download('punkt')

def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    tokens = word_tokenize(text)
    return " ".join(tokens)

df['cleaned_review'] = df['text'].apply(preprocess_text)
print(df[['text', 'cleaned_review']].head())
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
                                                text  \
0  This movie was horrible. If it had never been ...
1  The director infuses this film with false dept...
2  I don't get it! The teenage leads in "Horror S...
3  This is the fifth part of 'The Animatrix', a c...
4  I was very impressed with with this film which...

                                        cleaned_review
0  this movie was horrible if it had never been m...
1  the director infuses this film with false dept...
2  i dont get it the teenage leads in horror star...
3  this is the fifth part of the animatrix a coll...
4  i was very impressed with with this film which...
<ipython-input-16-3c43fad43701>:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df['cleaned_review'] = df['text'].apply(preprocess_text)
```

```
from nltk.corpus import stopwords
```

```
import nltk
nltk.download('stopwords')

stop_words = set(stopwords.words('english'))

def remove_stopwords(text):
```

```
    tokens = text.split()
    filtered_tokens = [word for word in tokens if word not in stop_words]
    return " ".join(filtered_tokens)

df['cleaned_review'] = df['cleaned_review'].apply(remove_stopwords)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
<ipython-input-19-3176d71aeb96>:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df['cleaned_review'] = df['cleaned_review'].apply(remove_stopwords)
```

```
from nltk.stem import WordNetLemmatizer

nltk.download('wordnet')
lemmatizer = WordNetLemmatizer()

def lemmatize_text(text):
    tokens = text.split()
    lemmatized_tokens = [lemmatizer.lemmatize(word) for word in tokens]
    return " ".join(lemmatized_tokens)

df['cleaned_review'] = df['cleaned_review'].apply(lemmatize_text)
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
<ipython-input-20-1d56ac0e0351>:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df['cleaned_review'] = df['cleaned_review'].apply(lemmatize_text)
```

```
print(df.columns)
```

```
Index(['text', 'cleaned_review'], dtype='object')
```

Lab #2.2 Sentiment Analysis

load a dataset named test.csv

```
import pandas as pd

df = pd.read_csv('/content/test.csv')
print("Dataset loaded successfully!")
print(df.head())
```

```
Dataset loaded successfully!
                                                text
0  This movie was horrible. If it had never been ...
1  The director infuses this film with false dept...
2  I don't get it! The teenage leads in "Horror S...
3  This is the fifth part of 'The Animatrix', a c...
4  I was very impressed with with this film which...
```

Pre-process the Review Data

- **Convert text to lowercase**
- **Remove special characters**
- **Tokenize**

```
import re
from nltk.tokenize import word_tokenize
import nltk
```

```
nltk.download('punkt')

def preprocess_text(text):
    if not isinstance(text, str):
        return ""
    text = text.lower()
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    tokens = word_tokenize(text)
    return " ".join(tokens)

df['cleaned_review'] = df['text'].apply(preprocess_text)
print(df[['text', 'cleaned_review']].head())
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
                                                text  \
0  This movie was horrible. If it had never been ...
1  The director infuses this film with false dept...
2  I don't get it! The teenage leads in "Horror S...
3  This is the fifth part of 'The Animatrix', a c...
4  I was very impressed with with this film which...

                                        cleaned_review
0  this movie was horrible if it had never been m...
1  the director infuses this film with false dept...
2  i dont get it the teenage leads in horror star...
3  this is the fifth part of the animatrix a coll...
4  i was very impressed with with this film which...
```

Sentiment Analysis Using Positive & Negative Word Lists

- **Download positive and negative word lists from Kaggle or use basic sets.**

```
positive_words = set(["good", "great", "excellent", "amazing", "wonderful", "best", "love"])
negative_words = set(["bad", "worst", "awful", "terrible", "poor", "hate"])
```

```
print("Positive words:", len(positive_words))
print("Positive words:", (positive_words))
print("Negative words:", len(negative_words))
print("Negative words:", (negative_words))
```

```
Positive words: 7
Positive words: {'excellent', 'amazing', 'great', 'love', 'best', 'wonderful', 'good'}
Negative words: 6
Negative words: {'hate', 'worst', 'terrible', 'bad', 'awful', 'poor'}
```

Classify Reviews as Positive, Negative, or Neutral

```
def analyze_sentiment(review):
    tokens = review.split()
    pos_count = sum(1 for word in tokens if word in positive_words)
    neg_count = sum(1 for word in tokens if word in negative_words)

    if pos_count > neg_count:
        return "Positive"
    elif neg_count > pos_count:
        return "Negative"
    else:
        return "Neutral"

df['sentiment'] = df['cleaned_review'].apply(analyze_sentiment)
print(df[['cleaned_review', 'sentiment']].head())
```

```
                                        cleaned_review sentiment
0  this movie was horrible if it had never been m...   Neutral
1  the director infuses this film with false dept...  Positive
2  i dont get it the teenage leads in horror star...  Positive
3  this is the fifth part of the animatrix a coll...  Positive
4  i was very impressed with with this film which...  Positive
```

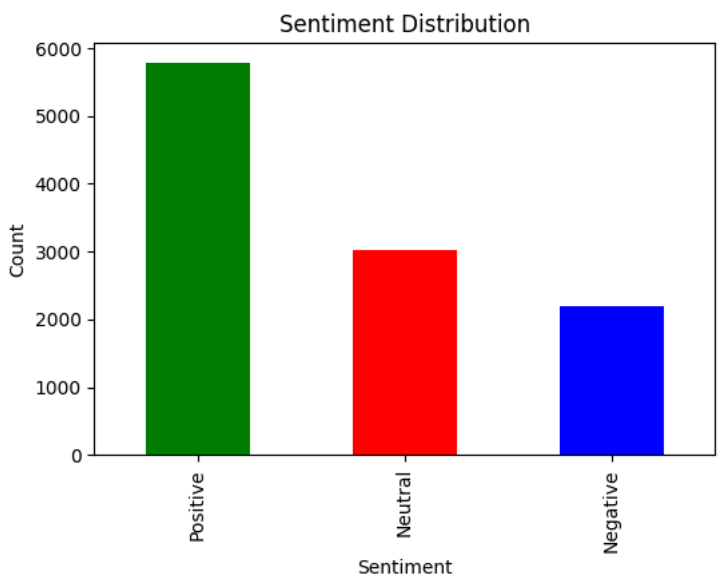
Review Results and Analyze Overall Sentiment Distribution

```
import matplotlib.pyplot as plt

sentiment_counts = df['sentiment'].value_counts()

plt.figure(figsize=(6,4))
sentiment_counts.plot(kind='bar', color=['green', 'red', 'blue'])
plt.title("Sentiment Distribution")
plt.xlabel("Sentiment")
plt.ylabel("Count")
plt.show()


print(sentiment_counts)
```




```
sentiment
Positive    5787
Neutral     3016
Negative    2198
Name: count, dtype: int64
```

Display Sample Reviews by Sentiment

```
for sentiment in ['Positive', 'Negative', 'Neutral']:
    print(f"\nSample {sentiment} Reviews:")
    print(df[df['sentiment'] == sentiment]['text'].head(3).tolist())
```

```
Sample Positive Reviews:
['The director infuses this film with false depth by repeating a gimmick throughout the film. EVERY single shot in this movie is 3 t

Sample Negative Reviews:
["This movie was horrible. If it had never been made the world would be a better place. Come on, a flying wagon? What were they thir

Sample Neutral Reviews:
["I just discovered this obscure '70s horror movie while browsing on YouTube. For a low-budget effort, it has plenty of compelling m
```

Start coding or generate with AI.

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.