# Part 1

```
In [ ]: import pandas as pd
```

```
In [ ]: df = pd.read_csv('imdb_movies.csv')
```

```
In [ ]: df.head()
```

Out[ ]:

| | names | date_x | score | genre | overview |
|---|---|---|---|---|---|
| **0** | Creed III | 03/02/2023 | 73.0 | Drama, Action | After dominating the boxing world, Adonis Cree... |
| **1** | Avatar: The Way of Water | 12/15/2022 | 78.0 | Science Fiction, Adventure, Action | Set more than a decade after the events of the... |
| **2** | The Super Mario Bros. Movie | 04/05/2023 | 76.0 | Animation, Adventure, Family, Fantasy, Comedy | While working underground to fix a water main,... |
| **3** | Mummies | 01/05/2023 | 70.0 | Animation, Comedy, Family, Adventure, Fantasy | Through a series of unfortunate events, three ... |
| **4** | Supercell | 03/17/2023 | 61.0 | Action | Good-hearted teenager William always lived in ... |

# Part 2

```
In [ ]: import nltk
        nltk.download('movie_reviews')
```

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

```
Out[ ]: True
```

```
In [ ]: from nltk.corpus import movie_reviews

        # View categories (positive/negative)
        print(movie_reviews.categories())
```

```
['neg', 'pos']
```

```
In [ ]: corpus = movie_reviews.raw()
```

```
In [ ]: print(corpus[:199])
```

```
plot : two teen couples go to a church party , drink and then drive .
they get into an accident .
one of the guys dies , but his girlfriend continues to see him in her life , and has
nightmares .
```

# Tokenization

```
In [ ]: import nltk
```

```
In [ ]: from nltk.tokenize import sent_tokenize, word_tokenize

        # Word Tokenize
        words = word_tokenize(corpus)

        # Sentence Tokenize
        sentences = sent_tokenize(corpus)
```

```
In [ ]: print(words[:10])
```

```
['plot', ':', 'two', 'teen', 'couples', 'go', 'to', 'a', 'church', 'party']
```

```
In [ ]: print(sentences[:2])
```

```
['plot : two teen couples go to a church party , drink and then drive .', 'they get
into an accident .']
```

# Stemming

```
In [ ]: from nltk.stem import PorterStemmer, WordNetLemmatizer

        # Stemming: Reduce words to their root form using Porter Stemmer
        stemmer = PorterStemmer()
        stemmed_words = [stemmer.stem(word) for word in words]
```

In [ ]:

```
print(stemmed_words[:10])
```

```
['plot', ':', 'two', 'teen', 'coupl', 'go', 'to', 'a', 'church', 'parti']
```

# Lemmatization

In [ ]:

```
from nltk.stem import PorterStemmer, WordNetLemmatizer

nltk.download('wordnet')

lemmatizer = WordNetLemmatizer()
lemmatized_words = [lemmatizer.lemmatize(word) for word in stemmed_words]
```

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

In [ ]:

```
print(lemmatized_words[:10])
```

```
['plot', ':', 'two', 'teen', 'coupl', 'go', 'to', 'a', 'church', 'parti']
```

# Stop Words

In [ ]:

```
from nltk.corpus import stopwords

nltk.download('stopwords')

stop_words = set(stopwords.words('english'))
filtered_words = [word for word in lemmatized_words if word.lower() not in stop_wor
```

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

In [ ]:

```
print(filtered_words[:10])
```

```
['plot', ':', 'two', 'teen', 'coupl', 'go', 'church', 'parti', ',', 'drink']
```

Each preprocessing step has its impact:

## Tokenization:

Splits the text into words and sentences. This step breaks down the text into its basic units, enabling further analysis at the word or sentence level.

## Stemming:

Reduces words to their root form. Stemming helps in reducing inflected words to their base or root form, which can help in normalization and reducing the vocabulary size.

## Lemmatization:

Further reduces stemmed words by considering their context. Lemmatization goes beyond stemming by considering the context of words and reducing them to their dictionary form or lemma, which can improve accuracy in some cases.

## Stop Word Removal:

Eliminates common words that may not be useful for analysis. Stop word removal helps in removing noise from the text and focusing on the most meaningful words for analysis.

Each preprocessing step plays a crucial role in preparing the text data for analysis, and the choice of which steps to include depends on the specific requirements of the NLP task at hand.