```
#installing nltk
import nltk
nltk.download('stopwords')
nltk.download('punkt_tab')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords 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!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
True
```

```
#import all the libraries we are going to use this experiment
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
```

```
#initialize
text = "This is a sample sentence showing off the stop words filtration and stem
stop_words = set(stopwords.words('english'))
ps = PorterStemmer()
lemmatizer = WordNetLemmatizer()
```

```
#tokenization
tokens = word_tokenize(text)
```

```
print("Original tokens:", tokens)
```

```
Original tokens: ['This', 'is', 'a', 'sample', 'sentence', 'showing', 'off', 'th
```

```
#stop word removel
filtered_tokens = [w for w in tokens if w.lower() not in stop_words]
```

```
print("After removing stop words:", filtered_tokens)
```

```
After removing stop words: ['sample', 'sentence', 'showing', 'stop', 'words', 'f
```

```
#stemming list
stemmed_tokens = [ps.stem(w) for w in filtered_tokens]
```

```
print("After stemming:", stemmed_tokens)
```

```
After stemming: ['sampl', 'sentenc', 'show', 'stop', 'word', 'filtrat', 'stem',
```

```
#lamatization list
lemmatized_tokens = [lemmatizer.lemmatize(w) for w in filtered_tokens]
```

```
print("After lemmatization:", lemmatized_tokens)
```

```
After lemmatization: ['sample', 'sentence', 'showing', 'stop', 'word', 'filtrati
```

```
# stemming preprocessing pipeline
def preprocess_text_stem(text):
    tokens = word_tokenize(text)
    filtered = [w for w in tokens if w.lower() not in stop_words]
    stemmed = [ps.stem(w) for w in filtered]
    return stemmed
```

```
# lemmatization preprocessing pipeline
def preprocess_text_lemma(text):
    tokens = word_tokenize(text)
    filtered = [w for w in tokens if w.lower() not in stop_words]
    lemmatized = [lemmatizer.lemmatize(w) for w in filtered]
    return lemmatized
```

```
test_texts = [
    "Programming languages are used by programmers to program",
    "The running runners were running fast"
]
```

```
#comparing stemming vs lemmatization use case
for text in test_texts:
    stem_result = preprocess_text_stem(text)
    lemma_result = preprocess_text_lemma(text)
    print(f"Original: '{text}'")
    print(f"Stemmed: {stem_result}")
    print(f"Lemmatized: {lemma_result}")
    print("-" * 50)
```

```
Original: 'Programming languages are used by programmers to program'
Stemmed: ['program', 'languag', 'use', 'programm', 'program']
Lemmatized: ['Programming', 'language', 'used', 'programmer', 'program']
--------------------------------------------------
Original: 'The running runners were running fast'
Stemmed: ['run', 'runner', 'run', 'fast']
Lemmatized: ['running', 'runner', 'running', 'fast']
--------------------------------------------------
```

```
#comparing stemming vs lemmatization
words = ["running", "flies", "better", "feet", "geese", "studies"]
print("Word\t\tStemmed\t\tLemmatized")
print("-" * 40)
for word in words:
```

```
    stemmed = ps.stem(word)
    lemmatized = lemmatizer.lemmatize(word)
    print(f"{word}\t\t{stemmed}\t\t{lemmatized}")
```

```
Word            Stemmed         Lemmatized
----------------------------------------
running         run             running
flies           fli             fly
better          better          better
feet            feet            foot
geese           gees            goose
studies         studi           study
```