```
import nltk
import string
import numpy as np
import pandas as pd

from nltk.corpus import stopwords, wordnet
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

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

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```
documents = [
    # Sports
    "The football team won the championship",
    "Cricket players trained hard for the match",
    "The athlete broke the world record",

    # Politics
    "The government passed a new law",
    "Elections were held across the country",
    "The minister addressed the parliament",

    # Health
    "Doctors recommend regular exercise",
    "A balanced diet improves health",
    "The hospital introduced new treatment",

    # Technology
    "Artificial intelligence is transforming industries",
    "The smartphone uses advanced technology",
    "Cybersecurity is important in modern systems",

    # Mixed
    "The player used technology to improve performance",
    "Government uses data for policy decisions",
    "Doctors use AI for disease detection",
    "Healthy lifestyle includes exercise and diet",
    "Technology helps hospitals improve healthcare",
    "Sports analytics uses machine learning"
]
```

```
df = pd.DataFrame({"Text": documents})
df.head()
```

	Text
0	The football team won the championship
1	Cricket players trained hard for the match
2	The athlete broke the world record
3	The government passed a new law
4	Elections were held across the country

Next steps: Generate code with df | New interactive sheet

```
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

def preprocess(text):
    text = text.lower()
```

```
    text = text.translate(str.maketrans('', '', string.punctuation))
    tokens = word_tokenize(text)
    tokens = [w for w in tokens if w not in stop_words]
    tokens = [lemmatizer.lemmatize(w) for w in tokens]
    return " ".join(tokens)
```

```
nltk.download('punkt_tab')
df["Clean_Text"] = df["Text"].apply(preprocess)
df.head()
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
```

	Text	Clean_Text
0	The football team won the championship	football team championship
1	Cricket players trained hard for the match	cricket player trained hard match
2	The athlete broke the world record	athlete broke world record
3	The government passed a new law	government passed new law
4	Elections were held across the country	election held across country

Next steps: Generate code with df | New interactive sheet

```
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(df["Clean_Text"])
```

```
cosine_sim = cosine_similarity(tfidf_matrix)
```

```
for i in range(5):
    for j in range(i+1, i+3):
        print(f"Doc {i} & Doc {j} Similarity: {cosine_sim[i][j]:.2f}")
```

```
Doc 0 & Doc 1 Similarity: 0.00
Doc 0 & Doc 2 Similarity: 0.00
Doc 1 & Doc 2 Similarity: 0.00
Doc 1 & Doc 3 Similarity: 0.00
Doc 2 & Doc 3 Similarity: 0.00
Doc 2 & Doc 4 Similarity: 0.00
Doc 3 & Doc 4 Similarity: 0.00
Doc 3 & Doc 5 Similarity: 0.00
Doc 4 & Doc 5 Similarity: 0.00
Doc 4 & Doc 6 Similarity: 0.00
```

```
def jaccard_similarity(doc1, doc2):
    set1 = set(doc1.split())
    set2 = set(doc2.split())
    return len(set1 & set2) / len(set1 | set2)
```

```
for i in range(5):
    for j in range(i+1, i+3):
        score = jaccard_similarity(df["Clean_Text"][i], df["Clean_Text"][j])
        print(f"Doc {i} & Doc {j} Jaccard: {score:.2f}")
```

```
Doc 0 & Doc 1 Jaccard: 0.00
Doc 0 & Doc 2 Jaccard: 0.00
Doc 1 & Doc 2 Jaccard: 0.00
Doc 1 & Doc 3 Jaccard: 0.00
Doc 2 & Doc 3 Jaccard: 0.00
Doc 2 & Doc 4 Jaccard: 0.00
Doc 3 & Doc 4 Jaccard: 0.00
Doc 3 & Doc 5 Jaccard: 0.00
Doc 4 & Doc 5 Jaccard: 0.00
Doc 4 & Doc 6 Jaccard: 0.00
```

```
def wordnet_similarity(word1, word2):
    synsets1 = wordnet.synsets(word1)
    synsets2 = wordnet.synsets(word2)
    if not synsets1 or not synsets2:
        return 0
    return synsets1[0].wup_similarity(synsets2[0])
```

```
pairs = [
    ("doctor", "physician"),
    ("football", "sport"),
    ("hospital", "clinic"),
    ("technology", "innovation"),
    ("diet", "nutrition"),
    ("government", "administration"),
    ("player", "athlete"),
    ("disease", "illness"),
    ("computer", "machine"),
    ("law", "rule")
]

for w1, w2 in pairs:
    print(w1, w2, wordnet_similarity(w1, w2))
```

```
doctor physician 1.0
football sport 0.8888888888888888
hospital clinic 0.11764705882352941
technology innovation 0.125
diet nutrition 0.3333333333333333
government administration 0.26666666666666666
player athlete 0.6666666666666666
disease illness 0.9473684210526315
computer machine 0.9411764705882353
law rule 0.3076923076923077
```

Start coding or generate with AI.