```
pip install numpy scikit-learn nltk
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.22.4)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.10.1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.1.0)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.3)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2022.10.31)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.65.0)
```

```
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
nltk.download('stopwords')
nltk.download('punkt')
```

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

```
def preprocess_text(text):
    stop_words = set(stopwords.words('english'))
    tokens = word_tokenize(text.lower())
    tokens = [token for token in tokens if token.isalnum() and token not in stop_words]
    return ' '.join(tokens)
```

```
def calculate_cosine_similarity(doc1, doc2):
    tfidf_vectorizer = TfidfVectorizer()
    tfidf_matrix = tfidf_vectorizer.fit_transform([doc1, doc2])
    cosine_sim = cosine_similarity(tfidf_matrix[0], tfidf_matrix[1])[0][0]
    return cosine_sim
```

```
document1 = input("Enter the first document: ")
document2 = input("Enter the second document: ")
```

```
Enter the first document: Hi I am Annie
Enter the second document: Hi I am Annie
```

```
processed_doc1 = preprocess_text(document1)
processed_doc2 = preprocess_text(document2)

similarity = calculate_cosine_similarity(processed_doc1, processed_doc2)
```

```
threshold = 0.8  # Adjust the threshold as per your requirements

if similarity > threshold:
   print("Plagiarism detected!")
else:
    print("No plagiarism detected.")
```

```
Plagiarism detected!
```

```
print('The percentage similarity is:',similarity*100)
```

```
The percentage similarity is: 100.00000000000003
```

✓ 0s completed at 12:18 AM ● ×