Ex4a: Information Retrieval using NLTK

```
import pandas as pd
import numpy as np
import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```
from google.colab import drive
drive.mount('/content/drive')
```

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

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

```
df = pd.read_csv("/content/Reviews.csv")
reviews = df[['Text']]
reviews.dropna(inplace=True)
reviews = reviews[:10000]
```

```
/tmp/ipython-input-1498711623.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ve
  reviews.dropna(inplace=True)
```

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

```
def preprocess(text):
    text = text.lower()
    text = re.sub(r'[^a-z\s]', '', text)
    tokens = word_tokenize(text)
    filtered_tokens = [word for word in tokens if word not in stop_words]
    return ' '.join(filtered_tokens)
```

```
reviews['Cleaned_Text'] = reviews['Text'].apply(preprocess)
```

```
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(reviews['Cleaned_Text'])
```

```
def search_reviews(query, k=5):
    cleaned_query = preprocess(query)
    query_vec = vectorizer.transform([cleaned_query])
    similarity = cosine_similarity(query_vec, tfidf_matrix)
    top_k_indices = similarity[0].argsort()[-k:][::-1]
    results = reviews.iloc[top_k_indices][['Text', 'Cleaned_Text']]
    return results
```

```
  print("Great product")
  print(search_reviews("great product"))
```

```
Great product
                                                   Text  \
8952  This is a good product with great price.  I re...
5617  I have ordered this product twice now and the ...
2557  I have very little to say about the product ex...
4471  This is a great product. Great flavors and ver...
```

```
3681  It is great! I like it alot. Great price too. ...

                                           Cleaned_Text
8952  good product great price received treat please...
5617  ordered product twice service product great wo...
2557  little say product except great product delive...
4471  great product great flavors fresh received qui...
3681  great like alot great price think delicious to...
```

```
print("Query: disappointed")
print(search_reviews("worst"))
```

```
Query: disappointed
                                                   Text  \
4592  This was the worst tasting tea, actually the w...
5974  I am big coffee lover.  This was some of the w...
9381  The worst!!! it is just plan awful bitter and ...
8672  The worst!!! it is just plan awful bitter and ...
4863  this gum is the worst i have ever purchased, p...

                                           Cleaned_Text
4592  worst tasting tea actually worst tasting anyth...
5974  big coffee lover worst coffee ever smells wond...
9381  worst plan awful bitter strong taste hazel nut...
8672  worst plan awful bitter strong taste hazel nut...
4863  gum worst ever purchased plain simple within t...
```