Ex4b: Information Retrieval using SpaCy

```
!pip install spacy scikit-learn pandas
!python -m spacy download en_core_web_sm
```

Show hidden output

```
import pandas as pd
df = pd.read_csv("/content/sample_data/Reviews.csv")
df = df.dropna(subset=["Text"])
df = df[['Text']].head(1000)
```

```
import spacy
nlp = spacy.load("en_core_web_sm")

def spacy_preprocess(text):
    doc = nlp(text.lower())
    tokens = [
        token.lemma_ for token in doc
        if token.is_alpha and not token.is_stop
    ]
    return ' '.join(tokens)
```

```
df['Cleaned_Text'] = df['Text'].apply(spacy_preprocess)
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

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

```
def process_query(query):
    query_cleaned = spacy_preprocess(query)
    query_vector = vectorizer.transform([query_cleaned])
    return query_vector
```

```
from sklearn.metrics.pairwise import cosine_similarity

def retrieve_reviews(query, k=5):
    query_vec = process_query(query)
    cosine_sim = cosine_similarity(query_vec, tfidf_matrix).flatten()

    top_k_idx = cosine_sim.argsort()[-k:][::-1]

    results = df.iloc[top_k_idx].copy()
    results['Similarity_Score'] = cosine_sim[top_k_idx]

    return results[['Text', 'Cleaned_Text', 'Similarity_Score']]
```

```
results = retrieve_reviews("great product", k=5)
print(results)
```

```
                                                 Text  \
42   I have McCann's Oatmeal every morning and by o...
181  This is an great product. The taste is great, ...
25   Product received is as advertised.<br /><br />...
934  I have 12 month olds and no time to write a gr...
661  I ordered this product two times now and have ...

                                         Cleaned_Text  Similarity_Score
42   mccann oatmeal morning order amazon able save ...          0.384441
181  great product taste great work exactly describ...          0.331609
25   product receive gp product strawberry ounce ba...          0.330008
934  month old time write great review like flavor ...          0.321406
661  order product time happy delivery product work...          0.320293
```

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

texts = ["This product is best", "This product is amazing"]
```

```
vectorizer = TfidfVectorizer()
tfidf = vectorizer.fit_transform(texts)

similarity = cosine_similarity(tfidf[0:1], tfidf[1:2])
print(similarity)
```

```
[[0.60297482]]
```