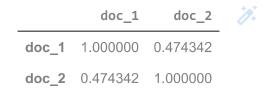
41443 - Anuj Mutha

```
doc_1 = "Data is the oil of the digital economy"
doc 2 = "Data is a new oil"
data = [doc_1, doc_2]
from sklearn.feature extraction.text import CountVectorizer
count vectorizer = CountVectorizer()
vector matrix = count vectorizer.fit transform(data)
vector matrix
    <2x8 sparse matrix of type '<class 'numpy.int64'>'
            with 11 stored elements in Compressed Sparse Row format>
tokens = count vectorizer.get feature names out()
tokens
    array(['data', 'digital', 'economy', 'is', 'new', 'of', 'oil', 'the'],
          dtype=object)
vector matrix.toarray()
    array([[1, 1, 1, 1, 0, 1, 1, 2],
           [1, 0, 0, 1, 1, 0, 1, 0]])
 Saving...
def create_dataframe(matrix, tokens):
    doc_names = [f'doc_{i+1}' for i, _ in enumerate(matrix)]
    df = pd.DataFrame(data=matrix, index=doc names, columns=tokens)
    return(df)
create_dataframe(vector_matrix.toarray(),tokens)
```

	data	digital	economy	is	new	of	oil	the	7
doc_1	1	1	1	1	0	1	1	2	
doc_2	1	0	0	1	1	0	1	0	

from sklearn.metrics.pairwise import cosine_similarity

cosine_similarity_matrix = cosine_similarity(vector_matrix)
create_dataframe(cosine_similarity_matrix,['doc_1','doc_2'])



from sklearn.feature extraction.text import TfidfVectorizer

Tfidf_vect = TfidfVectorizer()
vector_matrix = Tfidf_vect.fit_transform(data)

tokens = Tfidf_vect.get_feature_names_out()
create_dataframe(vector_matrix.toarray(),tokens)

	data	digital	economy	is	new	of	oil	the
doc_1	0.243777	0.34262	0.34262	0.243777	0.000000	0.34262	0.243777	0.68524
doc 2	0.448321	0.00000	0.00000	0.448321	0.630099	0.00000	0.448321	0.00000

cosine_similarity_matrix = cosine_similarity(vector_matrix)
create_dataframe(cosine_similarity_matrix,['doc_1','doc_2'])



Colab paid products - Cancel contracts here

✓ 0s completed at 1:36 PM

Saving... X