Exercise:) schoor. pealure extraction. [ex] import . The Rectarizer impord pardous as pol docs = (1 good movie", "not a good movie", "halfd not like", P'IPKe "t", "good one" tith = Tidy vectorizer. [min_db=2, max.db=0.5] ngram_range = (1,2)) beatures = thaff . fit fransform (docs) Print (teatures) of =pd, Dataframe (.features. todonse(), Columns = Effelf, get-features-names()) Print (db) · High = Therestorizer (min, d)=1, max. db=0.6, features: third fit - transform (docs) ngram ranger (1,2) print / features of = pd. Data frame (beatines. todanse ()) Columns = Hidb. get-leasure-hames ()) print(db) from skleam. netrics. pairwise Proport Whear-ternel. docit > features [0:1] doce = beatures (1:27 = thear-ternel (doct, docz)

Scanned with CamScanner

Natural Language Processing Lab Lab3. Computing Document Similarity using VSM

EXERCISE-1: Print TFIDF values

```
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd

docs = [
    "good movie", "not a good movie", "did not like",
    "i like it", "good one" ]

# using default tokenizer in TfidfVectorizer
tfidf = TfidfVectorizer(min_df=2, max_df=0.5, ngram_range=(1, 2))
features = tfidf.fit_transform(docs)
print(features)

# Pretty printing
df = pd.DataFrame(
    features.todense(),
    columns=tfidf.get_feature_names())
print(df)
```

EXERCISE-2:

(1)

1. Change the values of min_df and ngram_range and observe various outputs

EXERCISE-3: Compute Cosine Similarity between 2 Documents

```
from sklearn.metrics.pairwise import linear_kernel
# cosine score between 1st and 2nd doc
doc1 = features[0:1]
doc2 = features[1:2]
score = linear_kernel(doc1, doc2)
print(score)
# cosine score between 1st and all other docs
scores = linear_kernel(doc1, features)
print(scores
# Cosine Similarity for a new doc
query = "I like this good movie"
qfeature = tfidf.transform([query]
scores2 = linear_kernel(doc1, features)
print(scores2)
```

EXERCISE-4: Find Top-N similar documents

Question-1. Consider the following documents and compute TFIDF values

```
docs=["the house had a tiny little mouse",
"the cat saw the mouse",
"the mouse ran away from the house",
"the cat finally ate the mouse",
"the end of the mouse story"
```

Question-2. Compute cosine similarity between $3r^d$ document ("the mouse ran away from the house") with all other documents. Which is the most similar document?.

Question-3. Find Top-2 similar documents for the 3rd document based on Cosine similarity values.

DEPT OF DATA SCIENCE | BISHOP HEBER COLLEGE | TRICHY

DR. K. RAJKUMAR -

score = linear - kernel (doe 1, alter) print (scores) growy = 11 1 like. Hus good movie" apeature = Hidf . transform ([query]) Scor = Linear- Kernel (doc1, jeatures) print (scor) EXTH Guest :1 does - [" the house had a ling little mouse, " the cat saw the moure", If the mouse ran away from the house little cat finally are the movee", I the end of the mouse story! total = : they bectorios (min_db = 2, max_db=0.5) ngram rang = (1,2)) deatures = flidt. fit toansform (docs) popp of (features) doc1 = features [0:3] 1) = linear- kernel (doct, features) bush (8) Scores 2 = linear-bernel (doi 1, beatures) print (scores 2)