

**School of Computer Science and Engineering**

**Register Number: 18BCE0745**

**Name: Gourishetty Sreemanth**

**Code:**

import pandas as pd

from sklearn.feature\_extraction.text import TfidfTransformer

from sklearn.feature\_extraction.text import CountVectorizer

docs=["plot: two teen couples go to a church party, drink and then drive.",

"films adapted from comic books have had plenty of success , whether they're about superheroes ( batman , superman , spawn ) , or geared toward kids ( casper ) or the arthouse crowd ( ghost world ) , but there's never really been a comic book like from hell before .",

"every now and then a movie comes along from a suspect studio , with every indication that it will be a stinker , and to everybody's surprise ( perhaps even the studio ) the film becomes a critical darling . ",

"damn that y2k bug ."

]

#########################

print("TFIDF")

from sklearn.feature\_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()

vectors = vectorizer.fit\_transform(docs)

feature\_names = vectorizer.get\_feature\_names()

dense = vectors.todense()

denselist = dense.tolist()

df = pd.DataFrame(denselist, columns=feature\_names)

print(df)

print("BOW")

from sklearn.feature\_extraction.text import CountVectorizer

cv=CountVectorizer()

word\_count\_vector=cv.fit\_transform(docs)

word\_count\_vector.shape

print(word\_count\_vector.toarray())

print("IDF")

tfidf\_transformer=TfidfTransformer(smooth\_idf=True,use\_idf=True)

tfidf\_transformer.fit(word\_count\_vector)

# print idf values

df\_idf = pd.DataFrame(tfidf\_transformer.idf\_, index=cv.get\_feature\_names(),columns=["idf\_weights"])

# sort ascending

df\_idf.sort\_values(by=['idf\_weights'])

print(df\_idf)

print("TF")

#instantiate CountVectorizer()

cv=CountVectorizer()

# this steps generates word counts for the words in your docs

word\_count\_vector=cv.fit\_transform(docs)

word\_count\_vector.shape

pd = pd.DataFrame(word\_count\_vector.toarray(), columns = cv.get\_feature\_names())

print(pd)

count\_vector=cv.transform(docs)

print(count\_vector.toarray())

tf\_idf\_vector=tfidf\_transformer.transform(count\_vector)

print(tf\_idf\_vector.toarray())

feature\_names = cv.get\_feature\_names()

print(feature\_names)



