CLASS ASSIGNMENT-NLP

BY: ARCHITA BAJPAI
REG NO: 23BAI10066
CLASS SLOT: B11+B12+B13

[11] ✓ 0s

```
#BY: ARCHITA BAJPAI
#REGN NO: 23BAI10066
# Import TfidfVectorizer from scikit-learn library
from sklearn.feature_extraction.text import TfidfVectorizer
```

[12] ✓ 0s

```
# Create a list of the given strings
string = ["This is the first document.",
    "This document is the second document.",
    "And this is the third one.",
    "Is this the first document?"]
```

[8] ✓ 0s

```
# Initialize the TfidfVectorizer object
tfidf = TfidfVectorizer()

# Fit the vectorizer to the documents and transform them into TF-IDF matrix
result = tfidf.fit_transform(string)
```

[13] ✓ 0s

```
# Display IDF values for each word
print('\nidf values:')
# Loop through feature names and their corresponding IDF values
for ele1, ele2 in zip(tfidf.get_feature_names_out(), tfidf.idf_):
    print(ele1, ':', ele2)
```

```
idf values:
and : 1.916290731874155
document : 1.2231435513142097
first : 1.5108256237659907
is : 1.0
one : 1.916290731874155
second : 1.916290731874155
the : 1.0
third : 1.916290731874155
this : 1.0
```

```
# Display the vocabulary dictionary
print('\nWord indexes:')
# vocabulary_ maps each word to its column index in the TF-IDF matrix
print(tfidf.vocabulary_)

# Display the TF-IDF result in sparse matrix format
print('\ntf-idf value:')
# This shows the sparse matrix representation
print(result)
# Display the TF-IDF values in dense matrix form
print('\ntf-idf values in matrix form:')
# Values represent the TF-IDF score of each word in each document
print(result.toarray())
```

OUTPUT:

```
Word indexes:
{'this': 8, 'is': 3, 'the': 6, 'first': 2, 'document': 1, 'second': 5, 'and': 0, 'third': 7, 'one': 4}

tf-idf value:
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 21 stored elements and shape (4, 9)>
  Coords        Values
  (0, 8)        0.38408524091481483
  (0, 3)        0.38408524091481483
  (0, 6)        0.38408524091481483
  (0, 2)        0.5802858236844359
  (0, 1)        0.46979138557992045
  (1, 8)        0.281088674033753
  (1, 3)        0.281088674033753
  (1, 6)        0.281088674033753
  (1, 1)        0.6876235979836938
  (1, 5)        0.5386476208856763
  (2, 8)        0.267103787642168
  (2, 3)        0.267103787642168
  (2, 6)        0.267103787642168
  (2, 0)        0.511848512707169
  (2, 7)        0.511848512707169
  (2, 4)        0.511848512707169
  (3, 8)        0.38408524091481483
  (3, 3)        0.38408524091481483
  (3, 6)        0.38408524091481483
  (3, 2)        0.5802858236844359
  (3, 1)        0.46979138557992045
```

```
tf-idf values in matrix form:
[[0.         0.46979139 0.58028582 0.38408524 0.         0.
  0.38408524 0.         0.38408524]
 [0.         0.6876236  0.         0.28108867 0.         0.53864762
  0.28108867 0.         0.28108867]
 [0.51184851 0.         0.         0.26710379 0.51184851 0.
  0.26710379 0.51184851 0.26710379]
 [0.         0.46979139 0.58028582 0.38408524 0.         0.
  0.38408524 0.         0.38408524]]
```