NAME : Arul Kumar ARK

ROLL NO. : 225229103

Lab : 3

Computing Document Similarity using VSM

EXERCISE- 1

In [6]:
```
from sklearn.feature_extraction.text import TfidfVectorizer

import pandas as pd
```

In [7]:
```
docs = ["good movie", "not a good movie", "did not like","i like it","good one"]
```

In [8]:
```
print(docs)
```

```
['good movie', 'not a good movie', 'did not like', 'i like it', 'good one']
```

In [9]:
```
tfidf=TfidfVectorizer(min_df=2,max_df=0.5,ngram_range=(1,2))
```

In [11]:
```
features=tfidf.fit_transform(docs)
```

In [12]:
```
print(features)
```

```
  (0, 2)        0.7071067811865476
  (0, 0)        0.7071067811865476
  (1, 2)        0.5773502691896257
  (1, 0)        0.5773502691896257
  (1, 3)        0.5773502691896257
  (2, 3)        0.7071067811865476
  (2, 1)        0.7071067811865476
  (3, 1)        1.0
```

In [14]:
```
df=pd.DataFrame(features.todense(),columns=tfidf.get_feature_names())
```

In [15]:
```
print(df)
```

```
   good movie      like     movie       not
0    0.707107  0.000000  0.707107  0.000000
1    0.577350  0.000000  0.577350  0.577350
2    0.000000  0.707107  0.000000  0.707107
3    0.000000  1.000000  0.000000  0.000000
4    0.000000  0.000000  0.000000  0.000000
```

EXERCISE- 2

In [23]:
```
tfidf=TfidfVectorizer(min_df=2,max_df=0.8,ngram_range=(2,2))
features=tfidf.fit_transform(docs)
print(features)
```

```
  (0, 0)        1.0
  (1, 0)        1.0
```

EXERCISE- 3

In [24]:
```
from sklearn.metrics.pairwise import linear_kernel
```

In [26]:
```
docl = features[0:1]
doc2 = features[1:2]
score = linear_kernel(docl, doc2)
print(score)
```

```
[[1.]]
```

In [27]:
```
scores = linear_kernel(docl, features)
print (scores)
```

```
[[1. 1. 0. 0. 0.]]
```

In [32]:
```
query = "I like this good movie"
feature = tfidf.transform([query])
scores2 = linear_kernel(docl, features)
print (scores2)
```

```
[[1. 1. 0. 0. 0.]]
```

EXERCISE- 4

In [30]:
```
e 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"]
```

In [31]:
```
docs
```

Out[31]:
```
['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']
```

In [33]:
```
tfidf=TfidfVectorizer(min_df=2,max_df=0.5,ngram_range=(1,2))
features=tfidf.fit_transform(docs)
print(features)
```

```
  (0, 1)        0.7071067811865476
  (0, 3)        0.7071067811865476
  (1, 0)        0.7071067811865476
  (1, 2)        0.7071067811865476
  (2, 1)        0.7071067811865476
  (2, 3)        0.7071067811865476
  (3, 0)        0.7071067811865476
  (3, 2)        0.7071067811865476
```

In [34]:
```
scores_2 = linear_kernel(features[3], features)
```

In [35]:
```
scores_2
```

Out[35]:
```
array([[0., 1., 0., 1., 0.]])
```

In [38]:
```
scores_3 = linear_kernel(features[3], features[0:2])
```

In [39]:
```
scores_3
```

Out[39]:
```
array([[0., 1.]])
```

In [ ]:
```

```