In [1]:
```
documents = (
"The sky is blue",
"The sun is bright",
"The sun in the sky is bright",
"We can see the shining sun, the bright sun"
)
```

In [2]:
```
documents
```

Out[2]:
```
('The sky is blue',
 'The sun is bright',
 'The sun in the sky is bright',
 'We can see the shining sun, the bright sun')
```

In [ ]:
```

```

In [31]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

In [33]:
```
cv =CountVectorizer()
bow_vectorizer= cv.fit_transform(documents)
```

In [35]:
```
bow_vectorizer.todense()
```

Out[35]:
```
matrix([[1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0],
        [0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0],
        [0, 1, 0, 1, 1, 0, 0, 1, 1, 2, 0],
        [0, 1, 1, 0, 0, 1, 1, 0, 2, 2, 1]], dtype=int64)
```

In [45]:
```
cv.get_feature_names()
```

Out[45]:
```
['blue',
 'bright',
 'can',
 'in',
 'is',
 'see',
 'shining',
 'sky',
 'sun',
 'the',
 'we']
```

In [53]:
```
pd.DataFrame(bow_vectorizer.todense(),columns=cv.get_feature_names(),
             index=['1st sent', '2nd sent', '3rd sent','4th sent'])
```

Out[53]:

	blue	bright	can	in	is	see	shining	sky	sun	the	we
1st sent	1	0	0	0	1	0	0	1	0	1	0
2nd sent	0	1	0	0	1	0	0	0	1	1	0
3rd sent	0	1	0	1	1	0	0	1	1	2	0
4th sent	0	1	1	0	0	1	1	0	2	2	1

In [41]:
```
bow_vectorizer[0:1].toarray()
```

Out[41]:
```
array([[1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0]], dtype=int64)
```

In [42]:
```
print(bow_vectorizer[0:1])
```

```
  (0, 9)        1
  (0, 7)        1
  (0, 4)        1
  (0, 0)        1
```

In [ ]:
```

```

In [3]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [4]:
```
tfidf_vectorizer = TfidfVectorizer()
```

In [29]:
```

```

In [5]:
```
tfidf_matrix = tfidf_vectorizer.fit_transform(documents)
```

In [11]:
```
tfidf_matrix
```

Out[11]:
```
<4x11 sparse matrix of type '<class 'numpy.float64'>'
        with 21 stored elements in Compressed Sparse Row format>
```

In [30]:
```
tfidf_matrix.todense()   ##Converted to Dense Matrix
```

Out[30]:
```
matrix([[0.65919112, 0.        , 0.        , 0.        , 0.42075315,
         0.        , 0.        , 0.51971385, 0.        , 0.34399327,
         0.        ],
        [0.        , 0.52210862, 0.        , 0.        , 0.52210862,
         0.        , 0.        , 0.        , 0.52210862, 0.42685801,
         0.        ],
        [0.        , 0.3218464 , 0.        , 0.50423458, 0.3218464 ,
         0.        , 0.        , 0.39754433, 0.3218464 , 0.52626104,
         0.        ],
        [0.        , 0.23910199, 0.37459947, 0.        , 0.        ,
         0.37459947, 0.37459947, 0.        , 0.47820398, 0.39096309,
         0.37459947]])
```

In [8]:
```
tfidf_matrix.shape
```

Out[8]:
```
(4, 11)
```

In [52]:
```
pd.DataFrame(tfidf_matrix.todense(),columns=tfidf_vectorizer.get_feature_names(),
             index=['1st sent', '2nd sent', '3rd sent','4th sent'])
```

Out[52]:

	blue	bright	can	in	is	see	shining	sky	sun	the	
1st sent	0.659191	0.000000	0.000000	0.000000	0.420753	0.000000	0.000000	0.519714	0.000000	0.343993	0.00
2nd sent	0.000000	0.522109	0.000000	0.000000	0.522109	0.000000	0.000000	0.000000	0.522109	0.426858	0.00
3rd sent	0.000000	0.321846	0.000000	0.504235	0.321846	0.000000	0.000000	0.397544	0.321846	0.526261	0.00
4th sent	0.000000	0.239102	0.374599	0.000000	0.000000	0.374599	0.374599	0.000000	0.478204	0.390963	0.37

In [12]:
```
```

In [24]:
```
print(tfidf_matrix[0:1])
```

```
  (0, 0)        0.6591911178676787
  (0, 4)        0.42075315164463567
  (0, 7)        0.5197138488789809
  (0, 9)        0.3439932714296342
```

In [22]:
```
print(tfidf_matrix[0:2])
```

```
  (0, 0)        0.6591911178676787
  (0, 4)        0.42075315164463567
  (0, 7)        0.5197138488789809
  (0, 9)        0.3439932714296342
  (1, 1)        0.5221086219944969
  (1, 8)        0.5221086219944969
  (1, 4)        0.5221086219944969
  (1, 9)        0.42685800978431027
```

In [28]:
```
print(tfidf_matrix[0:3])
```

```
  (0, 0)        0.6591911178676787
  (0, 4)        0.42075315164463567
  (0, 7)        0.5197138488789809
  (0, 9)        0.3439932714296342
  (1, 1)        0.5221086219944969
  (1, 8)        0.5221086219944969
  (1, 4)        0.5221086219944969
  (1, 9)        0.42685800978431027
  (2, 3)        0.5042345768555538
  (2, 1)        0.32184639875982174
  (2, 8)        0.32184639875982174
  (2, 4)        0.32184639875982174
  (2, 7)        0.3975443320946988
  (2, 9)        0.5262610401109715
```

In [19]:
```
print(tfidf_matrix[0:1].toarray())
```

```
[[0.65919112 0.         0.         0.         0.42075315 0.
  0.         0.51971385 0.         0.34399327 0.         ]]
```

In [43]:
```
## Now we have the TF-IDF matrix (tfidf_matrix) for each document (the number of
## rows of the matrix) with 11 tf-idf terms (the number of columns from the matrix),
## we can calculate the Cosine Similarity between the first document("The sky is blue")
## with each of the other documents of the set
```

In [20]:
```
from sklearn.metrics.pairwise import cosine_similarity
```

In [ ]:
```
cosine_similarity(tfidf_matrix[0:1], tfidf_matrix)
```

In [ ]:
```
array([[ 1.        ,  0.36651513,  0.52305744,  0.13448867]])
```

In [ ]:
```
## The 1st line "The sky is blue" gets a score of 1 with itself | with 2nd line it
## gets a score of 0.36 (not similar) with 3rd line it gets a score of 0.52
```

In [17]:
```
import math
# This was already calculated on the previous step, so we just use the value
cos_sim = 0.52305744
angle_in_radians = math.acos(cos_sim)
math.degrees(angle_in_radians)
```

Out[17]:
```
58.462437107432784
```

In [ ]:
```

```