In [1]:
```
import pandas as pd
import numpy as np
import collections
import re
```

In [2]:
```
doc1 = 'Game of Thrones is an amazing tv series'
doc2 = 'Game of Thrones is the best tv series!'
doc3 = 'Game of Thrones is so great'
```

In [5]:
```
l_doc1 = re.sub(r"[^a-zA-Z0-9]"," ",doc1.lower()).split()
l_doc2 = re.sub(r"[^a-zA-Z0-9]"," ",doc2.lower()).split()
l_doc3 = re.sub(r"[^a-zA-Z0-9]"," ",doc3.lower()).split()
```

In [6]:
```
l=l_doc1
l.extend(l_doc2)
l.extend(l_doc3)
l
```

Out[6]:
```
['game',
 'of',
 'thrones',
 'is',
 'an',
 'amazing',
 'tv',
 'series',
 'game',
 'of',
 'thrones',
 'is',
 'the',
 'best',
 'tv',
 'series',
 'game',
 'of',
 'thrones',
 'is',
 'so',
 'great']
```

In [8]:
```
wordset = set(l)
```

In [9]:
```
wordset
```

Out[9]:
```
{'amazing',
 'an',
 'best',
 'game',
 'great',
 'is',
 'of',
 'series',
 'so',
 'the',
 'thrones',
 'tv'}
```

In [12]:
```
def calculateBOW(wordset,l_doc):
    tf_diz = dict.fromkeys(wordset,0)
    for word in l_doc:
        tf_diz[word] = l_doc.count(word)

    return tf_diz
```

In [13]:
```
bow1 = calculateBOW(wordset, l_doc1)
bow2 = calculateBOW(wordset, l_doc2)
bow3 = calculateBOW(wordset, l_doc3)
df_bow = pd.DataFrame([bow1, bow2, bow3])
df_bow.head()
```

Out[13]:

	of	great	best	series	an	thrones	amazing	the	game	so	tv	is
0	3	1	1	2	1	3	1	1	3	1	2	3
1	1	0	1	1	0	1	0	1	1	0	1	1
2	1	1	0	0	0	1	0	0	1	1	0	1

In [14]:
```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()
```

In [19]:
```
X = vectorizer.fit_transform([doc1,doc2,doc3])
df_bow_sklearn = pd.DataFrame(X.toarray(), columns=vectorizer.get_feature_names_out
df_bow_sklearn.head()
```

Out[19]:

	amazing	an	best	game	great	is	of	series	so	the	thrones	tv
0	1	1	0	1	0	1	1	1	0	0	1	1
1	0	0	1	1	0	1	1	1	0	1	1	1
2	0	0	0	1	1	1	1	0	1	0	1	0

In [20]:
```
print(vectorizer.get_feature_names_out())
```

```
['amazing' 'an' 'best' 'game' 'great' 'is' 'of' 'series' 'so' 'the'
 'thrones' 'tv']
```

In [21]:
```
import nltk
import re
import numpy as np
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\bonde\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

Out[21]:
```
True
```

In [22]:
```
text = """Game of Thrones is an amazing tv series
Game of Thrones is the best tv series!
Game of Thrones is so great"""
dataset = nltk.sent_tokenize(text)
for i in range(len(dataset)):
    dataset[i] = dataset[i].lower()
    dataset[i] = re.sub(r'\W',' ',dataset[i])
    dataset[i] = re.sub(r'\s+',' ',dataset[i])
```

In [23]:
```
print(dataset)
```

```
['game of thrones is an amazing tv series game of thrones is the best tv series ',
'game of thrones is so great']
```

In [25]:
```
word2count = {}
for data in dataset:
    words = nltk.word_tokenize(data)
    for word in words:
        if word not in word2count.keys():
            word2count[word]=1
        else:
            word2count[word]+=1
```

In [26]:
```
word2count
```

Out[26]:
```
{'game': 3,
 'of': 3,
 'thrones': 3,
 'is': 3,
 'an': 1,
 'amazing': 1,
 'tv': 2,
 'series': 2,
 'the': 1,
 'best': 1,
 'so': 1,
 'great': 1}
```

In [ ]:
```

```