Feature extracting

In [1]:
```
1  Text = "I am learning NLP"
```

In [2]:
```
1  import pandas as pd
2  pd.get_dummies(Text.split())
```

Out[2]:

	I	NLP	am	learning
0	1	0	0	0
1	0	0	1	0
2	0	0	0	1
3	0	1	0	0

In [3]:
```
1  text = ["i love NLP and i will learn NLP in 2 months"]
```

In [4]:
```
1  from sklearn.feature_extraction.text import CountVectorizer
2  vectorizer = CountVectorizer()
3  vectorizer.fit(text)
4  vector = vectorizer.transform(text)
```

In [5]:
```
1  print(vectorizer.vocabulary_)
2  print(vector.toarray())
```

```
{'love': 3, 'nlp': 5, 'and': 0, 'will': 6, 'learn': 2, 'in': 1, 'months': 4}
[[1 1 1 1 1 2 1]]
```

In [6]:
```
1  print(vector)
```

```
  (0, 0)        1
  (0, 1)        1
  (0, 2)        1
  (0, 3)        1
  (0, 4)        1
  (0, 5)        2
  (0, 6)        1
```

In [7]:
```
1  CountVectorizer?
```

In [8]:
```
1  df = pd.DataFrame(data=vector.toarray(), columns=vectorizer.get_feature_names
2  df
```

Out[8]:

	and	in	learn	love	months	nlp	will
0	1	1	1	1	1	2	1

In [9]:
```
1  text = 'I am learning NLP'
```

In [10]:
```python
from textblob import TextBlob
TextBlob(text).ngrams(1)
```

Out[10]: [WordList(['I']), WordList(['am']), WordList(['learning']), WordList(['NLP'])]

In [11]:
```python
TextBlob(text).ngrams(2)
```

Out[11]: [WordList(['I', 'am']),
          WordList(['am', 'learning']),
          WordList(['learning', 'NLP'])]

In [12]:
```python
TextBlob(text).ngrams(3)
```

Out[12]: [WordList(['I', 'am', 'learning']), WordList(['am', 'learning', 'NLP'])]

In [13]:
```python
TextBlob(text).ngrams(4)
```

Out[13]: [WordList(['I', 'am', 'learning', 'NLP'])]

In [14]:
```python
Text = ["The quick brown fox jumped over the lazy dog.", "The dog.", "The fox
```

In [16]:
```python
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
vectorizer.fit(Text)
print(vectorizer.vocabulary_)
print(vectorizer.idf_)
```

```
{'the': 7, 'quick': 6, 'brown': 0, 'fox': 2, 'jumped': 3, 'over': 5, 'lazy': 4,
'dog': 1}
[1.69314718 1.28768207 1.28768207 1.69314718 1.69314718 1.69314718
 1.69314718 1.          ]
```

In [ ]: