convert Text-Numeric vectors

import packages

In [25]:

```
import pandas as pd
import nltk
import re
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
```

In [26]:

```
# Read file
txt=open("novel.txt",'r')
text=txt.read()
```

In [27]:

```
# split
words=re.split(r'\W+',text)
words[:10]
```

Out[27]:

```
['',
 'One',
 'Morning',
 'when',
 'Gregor',
 'Samsa',
 'woke',
 'from',
 'troubled',
 'dreams']
```

```
In [28]: # Puctuations
         string.punctuation

Out[28]: '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```
In [29]: # Preprocessing
         striped=[re.sub(r'\w\s',"",w) for w in words]
         striped[:10]

Out[29]: ['',
          'One',
          'Morning',
          'when',
          'Gregor',
          'Samsa',
          'woke',
          'from',
          'troubled',
          'dreams']
```

```
In [30]: # Lower case
         words=[word.lower() for word in striped]
         words[:10]

Out[30]: ['',
          'one',
          'morning',
          'when',
          'gregor',
          'samsa',
          'woke',
          'from',
          'troubled',
          'dreams']
```

In [31]:
```
# Remove Stopwords
alstpw=stopwords.words('english')

words=[i for i in words if i not in alstpw]
words[:10]
```

Out[31]:
```
['',
 'one',
 'morning',
 'gregor',
 'samsa',
 'woke',
 'troubled',
 'dreams',
 'found',
 'transformed']
```

In [32]:
```
# Tokenize
nltk.word_tokenize(str(words))
```

Out[32]:
```
['[',
 '``',
 ',',
 "'one",
 "'",
 ',',
 "'morning",
 "'",
 ',',
 "'gregor",
 "'",
 ',',
 "'samsa",
 "'",
 ',',
 "'woke",
 "'",
 ',',
 "'troubled",
 "'"
```

In [33]:

```
# Stemming
st=PorterStemmer()
words=[st.stem(word) for word in words]
words=[i for i in words if i not in alstpw]

words[:10]
```

Out[33]:

```
['',
 'one',
 'morn',
 'gregor',
 'samsa',
 'woke',
 'troubl',
 'dream',
 'found',
 'transform']
```

In [34]:

```
# Lematization
lemmatizer = WordNetLemmatizer()
```

In [35]:

```
wordsl=[lemmatizer.lemmatize(word) for word in words]
words=[i for i in words if i not in alstpw]

wordsl[:10]
```

Out[35]:

```
['',
 'one',
 'morn',
 'gregor',
 'samsa',
 'woke',
 'troubl',
 'dream',
 'found',
 'transform']
```

```
In [36]: # Vectorizer
         vectorizer = CountVectorizer()
         count_matrix = vectorizer.fit_transform(wordsl)
         count_matrix
```

```
Out[36]: <11843x2156 sparse matrix of type '<class 'numpy.int64'>'
                 with 11703 stored elements in Compressed Sparse Row format>
```

```
In [37]: # TF-IDF
         vectorizer2 = TfidfVectorizer()
         count_matrix2 = vectorizer2.fit_transform(wordsl)
         count_array2 = count_matrix2.toarray()
         count_array2
```

```
Out[37]: array([[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]])
```