Implementing CountVectorizer and Tf IDF in python

CountVectorizer (Using for Bag of Words)

CountVectorizer is used to transform a given text into a vector on the basis of frequency (count) of each word that occurs in the entire text.

In [38]:
```
from sklearn.feature_extraction.text import CountVectorizer
# list of text documents
text = ["hello, my name is Neel and I am aspiring data Scientist."]
text1 = ["hello You are watching unfold data science"]
```

In [39]:
```
vectorizer = CountVectorizer()

# tokenize and build vocab
vectorizer.fit(text)
```

Out[39]:
```
CountVectorizer(analyzer='word', binary=False, decode_error='strict',
                dtype=<class 'numpy.int64'>, encoding='utf-8', input='conten
t',
                lowercase=True, max_df=1.0, max_features=None, min_df=1,
                ngram_range=(1, 1), preprocessor=None, stop_words=None,
                strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None, vocabulary=None)
```

In [40]:
```
print(vectorizer.vocabulary_)
```

```
{'hello': 4, 'my': 6, 'name': 7, 'is': 5, 'neel': 8, 'and': 1, 'am': 0, 'aspi
ring': 2, 'data': 3, 'scientist': 9}
```

In [46]:
```
from IPython.display import Image
Image(filename=r'D:\NEEL_FOLDER\Data Science\NLP\Word_Vector.png')
```

Out[46]:

0	1	2	3	4	5	6	7	8	9
am	and	aspiring	data	hello	is	my	name	neel	scientist

- The words are arranged in ascending order. For understanding the concept I have arranged the words in above image respective of their vectors.
- The vectorizer has also done some basic cleaning.

In [41]:
```
# encode document
newvector = vectorizer.transform(text1)

# summarize encoded vector
print(newvector.toarray())
```

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

Observation

- After fitting the vectors on 'text1', the outputs tells us that apart from 3rd & 4th index others have 0 values.
- In the 3rd & 4th index we have 'data' & 'hello', so it shows that this words are also present in 'text1'.
- 0 value means those words are not common in this 2 sentences.

Tf IDF

Tf-Idf is a statistical measure that evaluates how relevant a word is to a document in a collection of documnents.

In [47]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
# list of text documents
text = ["Neel is a aspiring data scientist in India","This is unfold data scie
nce","Data Science is a promising career"]
```

In [49]:
```
# create the transform
vectorizer = TfidfVectorizer()
```

In [50]:
```
# tokenize and build vocab
vectorizer.fit(text)
```

Out[50]:
```
TfidfVectorizer(analyzer='word', binary=False, decode_error='strict',
                dtype=<class 'numpy.float64'>, encoding='utf-8',
                input='content', lowercase=True, max_df=1.0, max_features=Non
e,
                min_df=1, ngram_range=(1, 1), norm='l2', preprocessor=None,
                smooth_idf=True, stop_words=None, strip_accents=None,
                sublinear_tf=False, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None, use_idf=True, vocabulary=None)
```

In [54]:
```
print(vectorizer.vocabulary_)
```

```
{'neel': 6, 'is': 5, 'aspiring': 0, 'data': 2, 'scientist': 9, 'in': 3, 'indi
a': 4, 'this': 10, 'unfold': 11, 'science': 8, 'promising': 7, 'career': 1}
```

In [74]:
```
from IPython.display import Image
Image(filename=r'D:\NEEL_FOLDER\Data Science\NLP\Tf_Idf.png')
```

Out[74]:

0	1	2	3	4	5	6	7	8	9	10	11
aspiring	career	data	in	india	is	neel	promising	science	scientist	this	unfold

In [51]:
```
#Focus on IDF VALUES
print(vectorizer.idf_)
```

```
[1.69314718 1.69314718 1.         1.69314718 1.69314718 1.
 1.69314718 1.69314718 1.28768207 1.69314718 1.69314718 1.69314718]
```

Observation

- According to idf the word 'data' & 'is' is appearing maximum time (their value is minimum), which means they are less relevant.

Now taking the 1st document out of the list 'text'.

In [75]:
```
text_as_input = text[2]
text_as_input
```

Out[75]:
```
'Data Science is a promising career'
```

In [76]:
```
vector = vectorizer.transform([text_as_input])
```

In [77]:
```
print(vector.toarray())
```

```
[[0.         0.55249005 0.32630952 0.         0.         0.32630952
  0.         0.55249005 0.42018292 0.         0.         0.        ]]
```

Observation

- When the vectorizer is fit on text_as_input what can be seen is some of the value are 0, which means those words are not in the document.
- The words with highest value are 'career' & 'promising', which means they are uniquely present in the documents.

Source : Unfold datascience youtube channel