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

In [2]:
```
df = pd.read_csv("train.csv")
df
```

Out[2]:

	id	title	author	text	label
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1
...	...	...	...	...	...
20795	20795	Rapper T.I.: Trump a ’Poster Child For White S...	Jerome Hudson	Rapper T. I. unloaded on black celebrities who...	0
20796	20796	N.F.L. Playoffs: Schedule, Matchups and Odds -...	Benjamin Hoffman	When the Green Bay Packers lost to the Washing...	0
20797	20797	Macy’s Is Said to Receive Takeover Approach by...	Michael J. de la Merced and Rachel Abrams	The Macy’s of today grew from the union of sev...	0
20798	20798	NATO, Russia To Hold Parallel Exercises In Bal...	Alex Ansary	NATO, Russia To Hold Parallel Exercises In Bal...	1
20799	20799	What Keeps the F-35 Alive	David Swanson	David Swanson is an author, activist, journa...	1

20800 rows × 5 columns

In [3]:
```
df.head()
```

Out[3]:

	id	title	author	text	label
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

In [4]:
```
eg = ""
eg = '. '.join(df['title'].head())
print(eg)
```

```
House Dem Aide: We Didn’t Even See Comey’s Letter Until Jason Chaffetz Tweeted It. FLYN
N: Hillary Clinton, Big Woman on Campus - Breitbart. Why the Truth Might Get You Fired.
15 Civilians Killed In Single US Airstrike Have Been Identified. Iranian woman jailed fo
r fictional unpublished story about woman stoned to death for adultery
```

```
In [5]: import nltk
        from nltk.corpus import stopwords
```

StopWords

```
In [6]: stopword = stopwords.words("english")
        print(stopword)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've",
"you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himse
lf', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 't
hem', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that',
"that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'h
ave', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'bu
t', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'abo
ut', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'bel
ow', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'fu
rther', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'b
oth', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'onl
y', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don',
"don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'are
n', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "had
n't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't",
'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'was
n', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

```
In [7]: eg = eg.lower()
        eg
```

```
Out[7]: 'house dem aide: we didn’t even see comey’s letter until jason chaffetz tweeted it. flyn
        n: hillary clinton, big woman on campus - breitbart. why the truth might get you fired.
        15 civilians killed in single us airstrike have been identified. iranian woman jailed fo
        r fictional unpublished story about woman stoned to death for adultery'
```

Word Tokenizer

```
In [8]: words = nltk.word_tokenize(eg)
        print(words)
```

```
['house', 'dem', 'aide', ':', 'we', 'didn', '’', 't', 'even', 'see', 'comey', '’', 's',
'letter', 'until', 'jason', 'chaffetz', 'tweeted', 'it', '.', 'flynn', ':', 'hillary',
'clinton', ',', 'big', 'woman', 'on', 'campus', '-', 'breitbart', '.', 'why', 'the', 'tr
uth', 'might', 'get', 'you', 'fired', '.', '15', 'civilians', 'killed', 'in', 'single',
'us', 'airstrike', 'have', 'been', 'identified', '.', 'iranian', 'woman', 'jailed', 'fo
r', 'fictional', 'unpublished', 'story', 'about', 'woman', 'stoned', 'to', 'death', 'fo
r', 'adultery']
```

Sentence Tokenizer

```
In [9]: words1 = nltk.sent_tokenize(eg)
        print(words1)
```

```
['house dem aide: we didn’t even see comey’s letter until jason chaffetz tweeted it.',
'flynn: hillary clinton, big woman on campus - breitbart.', 'why the truth might get you
fired.', '15 civilians killed in single us airstrike have been identified.', 'iranian wo
man jailed for fictional unpublished story about woman stoned to death for adultery']
```

Removal of Stop Words

In [10]:

```
without_stopword = [word for word in words if word not in stopword]
print(without_stopword)
```

```
['house', 'dem', 'aide', ':', '’', 'even', 'see', 'comey', '’', 'letter', 'jason', 'chaf
fetz', 'tweeted', '.', 'flynn', ':', 'hillary', 'clinton', ',', 'big', 'woman', 'campu
s', '-', 'breitbart', '.', 'truth', 'might', 'get', 'fired', '.', '15', 'civilians', 'ki
lled', 'single', 'us', 'airstrike', 'identified', '.', 'iranian', 'woman', 'jailed', 'fi
ctional', 'unpublished', 'story', 'woman', 'stoned', 'death', 'adultery']
```

Lemmatizer

In [11]:

```
from nltk.stem import WordNetLemmatizer
from nltk.stem import SnowballStemmer
```

In [12]:

```
lemmatizer = WordNetLemmatizer()
lemmatized_output = ([lemmatizer.lemmatize(w) for w in without_stopword])
print(lemmatized_output)
```

```
['house', 'dem', 'aide', ':', '’', 'even', 'see', 'comey', '’', 'letter', 'jason', 'chaf
fetz', 'tweeted', '.', 'flynn', ':', 'hillary', 'clinton', ',', 'big', 'woman', 'campu
s', '-', 'breitbart', '.', 'truth', 'might', 'get', 'fired', '.', '15', 'civilian', 'kil
led', 'single', 'u', 'airstrike', 'identified', '.', 'iranian', 'woman', 'jailed', 'fict
ional', 'unpublished', 'story', 'woman', 'stoned', 'death', 'adultery']
```

Removal of Punctuation

In [13]:

```
without_punctuation = []
for q in without_stopword:
    if(q.isalpha()):
        without_punctuation.append(q)

print(without_punctuation)
```

```
['house', 'dem', 'aide', 'even', 'see', 'comey', 'letter', 'jason', 'chaffetz', 'tweete
d', 'flynn', 'hillary', 'clinton', 'big', 'woman', 'campus', 'breitbart', 'truth', 'migh
t', 'get', 'fired', 'civilians', 'killed', 'single', 'us', 'airstrike', 'identified', 'i
ranian', 'woman', 'jailed', 'fictional', 'unpublished', 'story', 'woman', 'stoned', 'dea
th', 'adultery']
```

In [14]:

```
lemmatizer = WordNetLemmatizer()
lemmatized_output = ([lemmatizer.lemmatize(w) for w in without_punctuation])
print(lemmatized_output)
```

```
['house', 'dem', 'aide', 'even', 'see', 'comey', 'letter', 'jason', 'chaffetz', 'tweete
d', 'flynn', 'hillary', 'clinton', 'big', 'woman', 'campus', 'breitbart', 'truth', 'migh
t', 'get', 'fired', 'civilian', 'killed', 'single', 'u', 'airstrike', 'identified', 'ira
nian', 'woman', 'jailed', 'fictional', 'unpublished', 'story', 'woman', 'stoned', 'deat
h', 'adultery']
```

In [15]:

```
stemmed_words = []
stemmer = SnowballStemmer("english")
for word in without_punctuation:
    stemmed_words.append(stemmer.stem(word))

print(stemmed_words)
```

```
['hous', 'dem', 'aid', 'even', 'see', 'comey', 'letter', 'jason', 'chaffetz', 'tweet',
'flynn', 'hillari', 'clinton', 'big', 'woman', 'campus', 'breitbart', 'truth', 'might',
'get', 'fire', 'civilian', 'kill', 'singl', 'us', 'airstrik', 'identifi', 'iranian', 'wo
man', 'jail', 'fiction', 'unpublish', 'stori', 'woman', 'stone', 'death', 'adulteri']
```

POS Tagging

In [25]:

```
pos_tagged = nltk.pos_tag(without_punctuation)

print(pos_tagged)
```

```
[('house', 'NN'), ('dem', 'NN'), ('aide', 'RB'), ('even', 'RB'), ('see', 'VB'), ('come
y', 'JJ'), ('letter', 'NN'), ('jason', 'NN'), ('chaffetz', 'NN'), ('tweeted', 'VBD'),
('flynn', 'JJ'), ('hillary', 'JJ'), ('clinton', 'NN'), ('big', 'JJ'), ('woman', 'NN'),
('campus', 'NN'), ('breitbart', 'NN'), ('truth', 'NN'), ('might', 'MD'), ('get', 'VB'),
('fired', 'VBN'), ('civilians', 'NNS'), ('killed', 'VBN'), ('single', 'JJ'), ('us', 'PR
P'), ('airstrike', 'IN'), ('identified', 'VBN'), ('iranian', 'JJ'), ('woman', 'NN'), ('j
ailed', 'VBD'), ('fictional', 'JJ'), ('unpublished', 'JJ'), ('story', 'NN'), ('woman',
'NN'), ('stoned', 'VBD'), ('death', 'NN'), ('adultery', 'NN')]
```

TF-IDF

In [26]:

```
# import required module
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [31]:

```
# assign documents
d0 = 'Sachin was the GOAT of the previous generation'
d1 = 'Virat was the GOAT of this generation'
d2 = 'Anushka will be the GOAT of the next generation'

# merge documents into a single corpus
string = [d0, d1, d2]
string
```

Out[31]:

```
['Sachin was the GOAT of the previous generation',
 'Virat was the GOAT of this generation',
 'Anushka will be the GOAT of the next generation']
```

In [32]:

```
# create object
tfidf = TfidfVectorizer()

# get tf-df values
result = tfidf.fit_transform(string)
```

In [33]:

```
# get idf values
print('\nidf values:')
for ele1, ele2 in zip(tfidf.get_feature_names_out(), tfidf.idf_):
        print(ele1, ':', ele2)
```

```
idf values:
anushka : 1.6931471805599454
be : 1.6931471805599454
generation : 1.0
goat : 1.0
next : 1.6931471805599454
of : 1.0
previous : 1.6931471805599454
sachin : 1.6931471805599454
the : 1.0
this : 1.6931471805599454
virat : 1.6931471805599454
was : 1.2876820724517808
will : 1.6931471805599454
```

In [34]:

```
# get indexing
print('\nWord indexes:')
print(tfidf.vocabulary_)

# display tf-idf values
print('\ntf-idf value:')
print(result)

# in matrix form
print('\ntf-idf values in matrix form:')
print(result.toarray())
```

```
Word indexes:
{'sachin': 7, 'was': 11, 'the': 8, 'goat': 3, 'of': 5, 'previous': 6, 'generation': 2,
'virat': 10, 'this': 9, 'anushka': 0, 'will': 12, 'be': 1, 'next': 4}

tf-idf value:
  (0, 2)	0.26359985093596655
  (0, 6)	0.44631334440825365
  (0, 5)	0.26359985093596655
  (0, 3)	0.26359985093596655
  (0, 8)	0.5271997018719331
  (0, 11)	0.3394328023512059
  (0, 7)	0.44631334440825365
  (1, 9)	0.5016513317715935
  (1, 10)	0.5016513317715935
  (1, 2)	0.2962833577206743
  (1, 5)	0.2962833577206743
  (1, 3)	0.2962833577206743
  (1, 8)	0.2962833577206743
  (1, 11)	0.3815187681027303
  (2, 4)	0.39400039808922477
  (2, 1)	0.39400039808922477
  (2, 12)	0.39400039808922477
  (2, 0)	0.39400039808922477
  (2, 2)	0.23270298212286766
  (2, 5)	0.23270298212286766
  (2, 3)	0.23270298212286766
  (2, 8)	0.4654059642457353

tf-idf values in matrix form:
[[0.         0.         0.26359985 0.26359985 0.         0.26359985
  0.44631334 0.44631334 0.5271997  0.         0.         0.3394328
  0.        ]
 [0.         0.         0.29628336 0.29628336 0.         0.29628336
  0.         0.         0.29628336 0.50165133 0.50165133 0.38151877
  0.        ]
 [0.3940004  0.3940004  0.23270298 0.23270298 0.3940004  0.23270298
  0.         0.         0.46540596 0.         0.         0.
  0.3940004 ]]
```

```
In [ ]:
```