In [10]:
```
! pip install nltk -U
! pip install bs4 -U
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: nltk in c:\programdata\anaconda3\lib\site-packages
(3.9.1)
Requirement already satisfied: click in c:\programdata\anaconda3\lib\site-packages
(from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\programdata\anaconda3\lib\site-packages
(from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\programdata\anaconda3\lib\site-
packages (from nltk) (2024.9.11)
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\site-packages (f
rom nltk) (4.66.5)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-package
s (from click->nltk) (0.4.6)
Defaulting to user installation because normal site-packages is not writeable
Collecting bs4
  Downloading bs4-0.0.2-py2.py3-none-any.whl.metadata (411 bytes)
Requirement already satisfied: beautifulsoup4 in c:\programdata\anaconda3\lib\site-p
ackages (from bs4) (4.12.3)
Requirement already satisfied: soupsieve>1.2 in c:\programdata\anaconda3\lib\site-pa
ckages (from beautifulsoup4->bs4) (2.5)
Downloading bs4-0.0.2-py2.py3-none-any.whl (1.2 kB)
Installing collected packages: bs4
Successfully installed bs4-0.0.2
```

In [14]:
```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\dell\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\dell\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\dell\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\dell\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
```

Out[14]:
```
True
```

In [16]:
```
import nltk
```

In [7]:
```
para='Rajgad(literal meaning Ruling Fort) is a hill fort situated in the pune distr
```

In [9]:
```
print(para)
```

```
Rajgad(literal meaning Ruling Fort) is a hill fort situated in the pune district of
maharashtra,India.Formerly known as Murumdev,the fort was the capital of Maratha emp
ire under the rule of Chatrapati Shivaji Maharaj for almost 26 years,after which the
capital moved to the Raigad Fort.[1]Treasures discovered from an adjacent fort calle
d Torna were used to completely built and fortify the rajgad fort.
```

In [11]:
```
para.split()
```

Out[11]:
```
['Rajgad(literal',
 'meaning',
 'Ruling',
 'Fort)',
 'is',
 'a',
 'hill',
 'fort',
 'situated',
 'in',
 'the',
 'pune',
 'district',
 'of',
 'maharashtra,India.Formerly',
 'known',
 'as',
 'Murumdev,the',
 'fort',
 'was',
 'the',
 'capital',
 'of',
 'Maratha',
 'empire',
 'under',
 'the',
 'rule',
 'of',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'for',
 'almost',
 '26',
 'years,after',
 'which',
 'the',
 'capital',
 'moved',
 'to',
 'the',
 'Raigad',
 'Fort.[1]Treasures',
 'discovered',
 'from',
 'an',
 'adjacent',
 'fort',
 'called',
 'Torna',
 'were',
 'used',
 'to',
 'completely',
 'built',
```

```
 'and',
 'fortify',
 'the',
 'rajgad',
 'fort.']
```

In [13]:
```
from nltk.tokenize import sent_tokenize
from nltk.tokenize import word_tokenize
```

In [22]:
```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\dell\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[22]: True

In [26]:
```
import nltk
nltk.download('punkt_tab')
```

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

Out[26]: True

In [28]:
```
sent=sent_tokenize(para)
```

In [40]:
```
sent[1]
```

Out[40]:
```
'[1]Treasures discovered from an adjacent fort called Torna were used to completel
y built and fortify the rajgad fort.'
```

In [42]:
```
words=word_tokenize(para)
```

In [44]:
```
words
```

Out[44]:
```
['Rajgad',
 '(',
 'literal',
 'meaning',
 'Ruling',
 'Fort',
 ')',
 'is',
 'a',
 'hill',
 'fort',
 'situated',
 'in',
 'the',
 'pune',
 'district',
 'of',
 'maharashtra',
 ',',
 'India.Formerly',
 'known',
 'as',
 'Murumdev',
 ',',
 'the',
 'fort',
 'was',
 'the',
 'capital',
 'of',
 'Maratha',
 'empire',
 'under',
 'the',
 'rule',
 'of',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'for',
 'almost',
 '26',
 'years',
 ',',
 'after',
 'which',
 'the',
 'capital',
 'moved',
 'to',
 'the',
 'Raigad',
 'Fort',
 '.',
 '[',
 '1',
```

```
 ']',
 'Treasures',
 'discovered',
 'from',
 'an',
 'adjacent',
 'fort',
 'called',
 'Torna',
 'were',
 'used',
 'to',
 'completely',
 'built',
 'and',
 'fortify',
 'the',
 'rajgad',
 'fort',
 '.']
```

In [46]:
```
from nltk.corpus import stopwords
```

In [48]:
```
swords=stopwords.words('english')
```

In [50]:
```
swords
```

Out[50]:
```
['a',
 'about',
 'above',
 'after',
 'again',
 'against',
 'ain',
 'all',
 'am',
 'an',
 'and',
 'any',
 'are',
 'aren',
 "aren't",
 'as',
 'at',
 'be',
 'because',
 'been',
 'before',
 'being',
 'below',
 'between',
 'both',
 'but',
 'by',
 'can',
 'couldn',
 "couldn't",
 'd',
 'did',
 'didn',
 "didn't",
 'do',
 'does',
 'doesn',
 "doesn't",
 'doing',
 'don',
 "don't",
 'down',
 'during',
 'each',
 'few',
 'for',
 'from',
 'further',
 'had',
 'hadn',
 "hadn't",
 'has',
 'hasn',
 "hasn't",
 'have',
 'haven',
```

```
"haven't",
'having',
'he',
"he'd",
"he'll",
'her',
'here',
'hers',
'herself',
"he's",
'him',
'himself',
'his',
'how',
'i',
"i'd",
'if',
"i'll",
"i'm",
'in',
'into',
'is',
'isn',
"isn't",
'it',
"it'd",
"it'll",
"it's",
'its',
'itself',
"i've",
'just',
'll',
'm',
'ma',
'me',
'mightn',
"mightn't",
'more',
'most',
'mustn',
"mustn't",
'my',
'myself',
'needn',
"needn't",
'no',
'nor',
'not',
'now',
'o',
'of',
'off',
'on',
'once',
'only',
```

```
'or',
'other',
'our',
'ours',
'ourselves',
'out',
'over',
'own',
're',
's',
'same',
'shan',
"shan't",
'she',
"she'd",
"she'll",
"she's",
'should',
'shouldn',
"shouldn't",
"should've",
'so',
'some',
'such',
't',
'than',
'that',
"that'll",
'the',
'their',
'theirs',
'them',
'themselves',
'then',
'there',
'these',
'they',
"they'd",
"they'll",
"they're",
"they've",
'this',
'those',
'through',
'to',
'too',
'under',
'until',
'up',
've',
'very',
'was',
'wasn',
"wasn't",
'we',
"we'd",
```

```
"we'll",
"we're",
'were',
'weren',
"weren't",
"we've",
'what',
'when',
'where',
'which',
'while',
'who',
'whom',
'why',
'will',
'with',
'won',
"won't",
'wouldn',
"wouldn't",
'y',
'you',
"you'd",
"you'll",
'your',
"you're",
'yours',
'yourself',
'yourselves',
"you've"]
```

In [54]:
```
x=[word for word in words if word not in swords]
```

In [56]:
```
x
```

Out[56]:
```
['Rajgad',
 '(',
 'literal',
 'meaning',
 'Ruling',
 'Fort',
 ')',
 'hill',
 'fort',
 'situated',
 'pune',
 'district',
 'maharashtra',
 ',',
 'India.Formerly',
 'known',
 'Murumdev',
 ',',
 'fort',
 'capital',
 'Maratha',
 'empire',
 'rule',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'almost',
 '26',
 'years',
 ',',
 'capital',
 'moved',
 'Raigad',
 'Fort',
 '.',
 '[',
 '1',
 ']',
 'Treasures',
 'discovered',
 'adjacent',
 'fort',
 'called',
 'Torna',
 'used',
 'completely',
 'built',
 'fortify',
 'rajgad',
 'fort',
 '.']
```

In [62]:
```
x=[word for word in words if word.lower() not in swords]
```

In [64]:
```
x
```

Out[64]:
```
['Rajgad',
 '(',
 'literal',
 'meaning',
 'Ruling',
 'Fort',
 ')',
 'hill',
 'fort',
 'situated',
 'pune',
 'district',
 'maharashtra',
 ',',
 'India.Formerly',
 'known',
 'Murumdev',
 ',',
 'fort',
 'capital',
 'Maratha',
 'empire',
 'rule',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'almost',
 '26',
 'years',
 ',',
 'capital',
 'moved',
 'Raigad',
 'Fort',
 '.',
 '[',
 '1',
 ']',
 'Treasures',
 'discovered',
 'adjacent',
 'fort',
 'called',
 'Torna',
 'used',
 'completely',
 'built',
 'fortify',
 'rajgad',
 'fort',
 '.']
```

In [68]:
```
from nltk.stem import PorterStemmer
```

In [70]:
```
ps=PorterStemmer()
```

In [72]:
```
ps.stem('working')
```

Out[72]:
```
'work'
```

In [81]:
```
y=[ps.stem(word) for word in x]
```

In [83]:
```
y
```

Out[83]:
```
['rajgad',
 '(',
 'liter',
 'mean',
 'rule',
 'fort',
 ')',
 'hill',
 'fort',
 'situat',
 'pune',
 'district',
 'maharashtra',
 ',',
 'india.formerli',
 'known',
 'murumdev',
 ',',
 'fort',
 'capit',
 'maratha',
 'empir',
 'rule',
 'chatrapati',
 'shivaji',
 'maharaj',
 'almost',
 '26',
 'year',
 ',',
 'capit',
 'move',
 'raigad',
 'fort',
 '.',
 '[',
 '1',
 ']',
 'treasur',
 'discov',
 'adjac',
 'fort',
 'call',
 'torna',
 'use',
 'complet',
 'built',
 'fortifi',
 'rajgad',
 'fort',
 '.']
```

In [87]:
```
from nltk.stem import WordNetLemmatizer
```

In [89]:
```
wnl=WordNetLemmatizer()
```

In [91]:
```
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\dell\AppData\Roaming\nltk_data...
```

Out[91]: True

In [93]:
```
wnl.lemmatize('working',pos='v')
#a-adjective
#n-noun
#r-adverb
```

Out[93]: 'work'

In [95]:
```
print(ps.stem('went'))
print(wnl.lemmatize('went',pos='v'))
```

```
went
go
```

In [97]:
```
z=[wnl.lemmatize(word,pos='v') for word in x]
```

In [99]:
```
z
```

Out[99]:
```
['Rajgad',
 '(',
 'literal',
 'mean',
 'Ruling',
 'Fort',
 ')',
 'hill',
 'fort',
 'situate',
 'pune',
 'district',
 'maharashtra',
 ',',
 'India.Formerly',
 'know',
 'Murumdev',
 ',',
 'fort',
 'capital',
 'Maratha',
 'empire',
 'rule',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'almost',
 '26',
 'years',
 ',',
 'capital',
 'move',
 'Raigad',
 'Fort',
 '.',
 '[',
 '1',
 ']',
 'Treasures',
 'discover',
 'adjacent',
 'fort',
 'call',
 'Torna',
 'use',
 'completely',
 'build',
 'fortify',
 'rajgad',
 'fort',
 '.']
```

In [101…
```
import string
```

In [103…
```
string.punctuation
```

Out[103…

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

In [105…

```
t=[word for word in words if word not in string.punctuation]
```

In [107…

```
t
```

Out[107…

```
['Rajgad',
 'literal',
 'meaning',
 'Ruling',
 'Fort',
 'is',
 'a',
 'hill',
 'fort',
 'situated',
 'in',
 'the',
 'pune',
 'district',
 'of',
 'maharashtra',
 'India.Formerly',
 'known',
 'as',
 'Murumdev',
 'the',
 'fort',
 'was',
 'the',
 'capital',
 'of',
 'Maratha',
 'empire',
 'under',
 'the',
 'rule',
 'of',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'for',
 'almost',
 '26',
 'years',
 'after',
 'which',
 'the',
 'capital',
 'moved',
 'to',
 'the',
 'Raigad',
 'Fort',
 '1',
 'Treasures',
 'discovered',
 'from',
 'an',
 'adjacent',
 'fort',
 'called',
```

```
 'Torna',
 'were',
 'used',
 'to',
 'completely',
 'built',
 'and',
 'fortify',
 'the',
 'rajgad',
 'fort']
```

In [111…
```
from nltk import pos_tag
```

In [115…
```
import nltk
nltk.download('averaged_perceptron_tagger_eng')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     C:\Users\dell\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger_eng.zip.
```

Out[115…
```
True
```

In [117…
```
pos_tag(t)
```

Out[117…

```
[('Rajgad', 'NNP'),
 ('literal', 'JJ'),
 ('meaning', 'NN'),
 ('Ruling', 'NNP'),
 ('Fort', 'NNP'),
 ('is', 'VBZ'),
 ('a', 'DT'),
 ('hill', 'NN'),
 ('fort', 'NN'),
 ('situated', 'VBN'),
 ('in', 'IN'),
 ('the', 'DT'),
 ('pune', 'JJ'),
 ('district', 'NN'),
 ('of', 'IN'),
 ('maharashtra', 'JJ'),
 ('India.Formerly', 'NNP'),
 ('known', 'VBN'),
 ('as', 'IN'),
 ('Murumdev', 'NNP'),
 ('the', 'DT'),
 ('fort', 'NN'),
 ('was', 'VBD'),
 ('the', 'DT'),
 ('capital', 'NN'),
 ('of', 'IN'),
 ('Maratha', 'NNP'),
 ('empire', 'NN'),
 ('under', 'IN'),
 ('the', 'DT'),
 ('rule', 'NN'),
 ('of', 'IN'),
 ('Chatrapati', 'NNP'),
 ('Shivaji', 'NNP'),
 ('Maharaj', 'NNP'),
 ('for', 'IN'),
 ('almost', 'RB'),
 ('26', 'CD'),
 ('years', 'NNS'),
 ('after', 'IN'),
 ('which', 'WDT'),
 ('the', 'DT'),
 ('capital', 'NN'),
 ('moved', 'VBD'),
 ('to', 'TO'),
 ('the', 'DT'),
 ('Raigad', 'NNP'),
 ('Fort', 'NNP'),
 ('1', 'CD'),
 ('Treasures', 'NNS'),
 ('discovered', 'VBN'),
 ('from', 'IN'),
 ('an', 'DT'),
 ('adjacent', 'JJ'),
 ('fort', 'NN'),
 ('called', 'VBN'),
```

```
 ('Torna', 'NNP'),
 ('were', 'VBD'),
 ('used', 'VBN'),
 ('to', 'TO'),
 ('completely', 'RB'),
 ('built', 'VBN'),
 ('and', 'CC'),
 ('fortify', 'VB'),
 ('the', 'DT'),
 ('rajgad', 'NN'),
 ('fort', 'NN')]
```

In [119…
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [121…
```
tfidf=TfidfVectorizer()
```

In [123…
```
v=tfidf.fit_transform(t)
```

In [125…
```
v.shape
```

Out[125… (67, 50)

In [129…
```
import pandas as pd
pd.DataFrame(v)
```

Out[129…

	0
0	(0, 35)\t1.0
1	(0, 25)\t1.0
2	(0, 29)\t1.0
3	(0, 37)\t1.0
4	(0, 17)\t1.0
…	…
62	(0, 5)\t1.0
63	(0, 18)\t1.0
64	(0, 40)\t1.0
65	(0, 35)\t1.0
66	(0, 17)\t1.0

67 rows × 1 columns

In [ ]: