In [ ]:
```
filename = 'novel.txt'
file = open(filename, 'rt')
text = file.read()
print(text)
#file.close()
```

In [ ]:
```
! pip install nltk
```

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

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

#text1 = "Python is a programming language. It is a interprested language, used for Data Anal
sent_tokens = sent_tokenize(text)
sent_tokens
```

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

#example_sent = "Python is a programming language. It is a interprested language, used for Da
word_tokens = word_tokenize(text)

word_tokens
```

In [ ]:
```
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
print(stop_words)
```

In [ ]:
```
filtered_sentence = []

for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)
print(filtered_sentence)
```

In [ ]:
```
new_stopwords = [".",",","/",":","<","=","!","-","_",";","''","``","'","?"]
```

In [ ]:
```
stop_words.extend(new_stopwords)
```

In [ ]:
```
filtered_sentence = []

for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)
print(filtered_sentence)
```

In [ ]:
```
from nltk.probability import FreqDist
fdist = FreqDist(filtered_sentence)
print(fdist)
```

In [ ]:
```
import matplotlib.pyplot as plt
fdist.plot(30,cumulative=False)
plt.show()
```

In [ ]:
```
fdist.most_common(20)
```

In [ ]:
```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()


words = word_tokenize(text)
```

```
print(words)

print('\n')

for w in words:
    print(ps.stem(w))
```

In [ ]:
```
import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()

words = word_tokenize(text)

print(words)

print('\n')

for w in words:
    print(lemmatizer.lemmatize(w))
```

Use SKLearn for converting Text-Numeric vectors using TF-IDF model

In [ ]:
```
import sklearn
from sklearn.feature_extraction.text import CountVectorizer
import pandas as pd
import numpy

import warnings
warnings.filterwarnings('ignore')
```

In [ ]:
```
corpus = ["Machine learning is super fun",
"Python is super, super cool",
"Statistics is cool, too",
"Data science is fun",
"Python is great for machine learning",
"I like football",
"Football is great to watch"]



vectorizer = CountVectorizer(stop_words='english', ngram_range=(1,2), max_df=0.8, min_df=0.2,
dtm = vectorizer.fit_transform(corpus)
pd.DataFrame(dtm.toarray(),index=corpus,columns=vectorizer.get_feature_names()).head(5)
```

In [ ]:
```
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.pipeline import Pipeline

pipe = Pipeline([('count', CountVectorizer()),('tfid', TfidfTransformer())]).fit(corpus)

pipe['count'].transform(corpus).toarray()
```

In [ ]:
```
pipe['tfid'].idf_
```

In [ ]:
```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(corpus)
X
```

In [ ]:
```
vectorizer.get_feature_names_out()
```

In [ ]:
```
print(X.shape)
```

In [ ]: