In [149…

```
from sklearn.feature_extraction.text  import CountVectorizer
```

In [150…

```
import pandas as pd
```

In [151…

```
from sklearn.feature_extraction.text  import CountVectorizer
```

In [152…

```
from sklearn.naive_bayes import BernoulliNB,MultinomialNB
```

In [153…

```
df = pd.read_csv("sentiment_analysis.csv")
```

In [154…

```
df.head()
```

Out[154…

	Sentence	Sentiment
0	The GeoSolutions technology will leverage Bene...	positive
1	$ESI on lows, down $1.50 to $2.50 BK a real po...	negative
2	For the last quarter of 2010 , Componenta 's n...	positive
3	According to the Finnish-Russian Chamber of Co...	neutral
4	The Swedish buyout firm has sold its remaining...	neutral

In [155…

```
df.isna()
```

Out[155…

	Sentence	Sentiment
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
...	...	...
5837	False	False
5838	False	False
5839	False	False
5840	False	False
5841	False	False

5842 rows × 2 columns

In [156…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5842 entries, 0 to 5841
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Sentence   5842 non-null   object
 1   Sentiment  5842 non-null   object
dtypes: object(2)
memory usage: 91.4+ KB
```

In [157…

```
vector1 = CountVectorizer(binary = True)
```

In [158…

```
vector2 =  CountVectorizer(binary = False)
```

In [159…

```
text = df['Sentence']
label = df['Sentiment']
```

In [160…

```
text
```

Out[160…

```
0       The GeoSolutions technology will leverage Bene...
1       $ESI on lows, down $1.50 to $2.50 BK a real po...
2       For the last quarter of 2010 , Componenta 's n...
3       According to the Finnish-Russian Chamber of Co...
4       The Swedish buyout firm has sold its remaining...
                              ...                        
5837    RISING costs have forced packaging producer Hu...
5838    Nordic Walking was first used as a summer trai...
5839    According shipping company Viking Line , the E...
5840    In the building and home improvement trade , s...
5841    HELSINKI AFX - KCI Konecranes said it has won ...
Name: Sentence, Length: 5842, dtype: object
```

In [161…

```
text.shape
```

Out[161…

```
(5842,)
```

In [162…

```
label
```

Out[162…

```
0       positive
1       negative
2       positive
3        neutral
4        neutral
          ...   
5837    negative
5838     neutral
5839     neutral
5840     neutral
5841    positive
Name: Sentiment, Length: 5842, dtype: object
```

In [163…

```
import nltk
from nltk.corpus import stopwords
```

In [164…

```python
# Download stop words list
nltk.download('stopwords')
nltk.download('punkt')  # Download the tokenizer model
stop_words = set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\PC-18\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\PC-18\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [165…

```python
stop_words
```

```
Out[165…  {'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',
'her',
'here',
'hers',
'herself',
'him',
'himself',
'his',
'how',
'i',
'if',
'in',
'into',
'is',
'isn',
"isn't",
'it',
"it's",
'its',
'itself',
'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's",
                   'should',
                   "should've",
                   'shouldn',
                   "shouldn't",
                   'so',
                   'some',
                   'such',
                   't',
                   'than',
                   'that',
                   "that'll",
                   'the',
                   'their',
                   'theirs',
                   'them',
                   'themselves',
                   'then',
                   'there',
                   'these',
                   'they',
                   'this',
                   'those',
                   'through',
                   'to',
                   'too',
                   'under',
                   'until',
                   'up',
                   've',
                   'very',
                   'was',
                   'wasn',
                   "wasn't",
                   'we',
                   'were',
                   'weren',
                   "weren't",
                   'what',
                   'when',
                   'where',
                   'which',
                   'while',
                   'who',
                   'whom',
                   'why',
                   'will',
                   'with',
                   'won',
                   "won't",
                   'wouldn',
 "wouldn't",
 'y',
 'you',
 "you'd",
 "you'll",
 "you're",
 "you've",
 'your',
 'yours',
 'yourself',
 'yourselves'}
```

In [166…

```
def preprocess_text(Sentence):
    Sentence = Sentence.lower()
    tokens = nltk.word_tokenize(Sentence)
    filtered_tokens = [word for word in tokens if word not in stop_words]
    return ' '.join(filtered_tokens)
```

In [167…

```
# Apply preprocessing to all texts
text = text.apply(preprocess_text)
```

In [168…

```
text
```

Out[168…

```
0       geosolutions technology leverage benefon 's gp...
1          $ esi lows , $ 1.50 $ 2.50 bk real possibility
2       last quarter 2010 , componenta 's net sales do...
3       according finnish-russian chamber commerce , m...
4       swedish buyout firm sold remaining 22.4 percen...
                              ...
5837    rising costs forced packaging producer huhtama...
5838    nordic walking first used summer training meth...
5839    according shipping company viking line , eu de...
5840    building home improvement trade , sales decrea...
5841    helsinki afx - kci konecranes said order four ...
Name: Sentence, Length: 5842, dtype: object
```

In [169…

```
X1 = vector1.fit_transform(text)
```

In [170…

```
X1
```

Out[170…

```
<Compressed Sparse Row sparse matrix of dtype 'int64'
        with 69381 stored elements and shape (5842, 11289)>
```

In [171…

```
X1.shape
```

Out[171…

```
(5842, 11289)
```

In [172…

```
X2 = vector2.fit_transform(text)
```

In [173…

```
X2.shape
```

Out[173…

```
(5842, 11289)
```

In [174…

```
y = label
```

In [175…
```
y.shape
```
Out[175…
```
(5842,)
```
In [176…
```
from sklearn.model_selection import train_test_split
```
In [177…
```
xtrain1,xtest1,ytrain,ytest = train_test_split(X1,y,test_size = 0.25,random_state =
```
In [178…
```
xtrain2,xtest2,ytrain,ytest = train_test_split(X2,y,test_size = 0.25,random_state =
```
In [179…
```
bnb = BernoulliNB()
```
In [180…
```
mnb = MultinomialNB()
```
In [181…
```
bnb.fit(xtrain1,ytrain)
```
Out[181…
```
▾ BernoulliNB
BernoulliNB()
```
In [182…
```
mnb.fit(xtrain2,ytrain)
```
Out[182…
```
▾ MultinomialNB
MultinomialNB()
```
In [183…
```
y_pred1 = bnb.predict(xtest1)
```
In [184…
```
y_pred2 = mnb.predict(xtest2)
```
In [185…
```
from sklearn.metrics import accuracy_score
```
In [186…
```
accuracy_score(ytest,y_pred1)
```
Out[186…
```
0.6680355920602327
```
In [187…
```
accuracy_score(ytest,y_pred2)
```
Out[187…
```
0.6598220396988365
```
In [188…
```
from sklearn.feature_extraction.text import TfidfVectorizer
```
In [189…
```
vector3 = TfidfVectorizer()
vector3
```

Out[189…

```
▾ TfidfVectorizer
TfidfVectorizer()
```

In [190…

```
X3 = vector3.fit_transform(text)
```

In [191…

```
xtrain,xtest,ytrain,ytest = train_test_split(X3,y,test_size = 0.25,random_state = 1
```

In [192…

```
model3=MultinomialNB()
```

In [193…

```
model3.fit(xtrain,ytrain)
```

Out[193…

```
▾ MultinomialNB
MultinomialNB()
```

In [194…

```
ypred = model3.predict(xtest)
```

In [195…

```
from sklearn.metrics import accuracy_score
```

In [196…

```
accuracy_score(ytest,ypred)
```

Out[196…

```
0.6584531143052703
```

The highest accuracy score is 0.66803 as compare to other vectorizer technique using BernoulliNB() with CountVectorizer after removing the default stop_words from the model.

In [ ]:

```

```