Sentiment Analysis

In [1]:
```
# importing libraries
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from string import punctuation
import contractions
import warnings
warnings.filterwarnings('ignore')
from unidecode import unidecode
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
import pandas as pd
from autocorrect import Speller
```

In [2]:
```
data = pd.read_csv('Sentimental_Analysis.csv')
data.head(2)
```

Out[2]:

	ï»¿text	label
0	I grew up (b. 1965) watching and loving the Th...	0
1	When I put this movie in my DVD player, and sa...	0

In [3]:
```
data = data.rename(columns = {data.columns[0]:'text'})
data.head()
```

Out[3]:

	text	label
0	I grew up (b. 1965) watching and loving the Th...	0
1	When I put this movie in my DVD player, and sa...	0
2	Why do people who do not know what a particula...	0
3	Even though I have great interest in Biblical ...	0
4	Im a die hard Dads Army fan and nothing will e...	1

In [ ]:
```
```

In [4]:
```
# 1. remove whitespaces blanklines
# 2. contraction mapping
# 3. cleaning - tokenization,stopwords removal,punctuations removal, numerical fo
# 4. autocorrection
# 5. handling accented characters
# 6. lemmatization

# 1. remove whitespaces blanklines
def remove_spaces(data):
    clean_text = data.replace("\\n"," ").replace("\n",' ').replace("\t"," ").repl
    return clean_text

# 2. contraction mapping
def expand_text(data):
    expanded_text = contractions.fix(data)
    return expanded_text

# 3. cleaning
stopword_list = stopwords.words("english")
stopword_list.remove("no")
stopword_list.remove("nor")
stopword_list.remove("not")
def clean_text(data,stopword_list):
    """tokenization,stopwords removal,normalization,punctuations removal, numeric
    tokens = word_tokenize(data)
    cleantext = [word.lower() for word in tokens if (word not in stopword_list) a
    return cleantext


# 4. autocorrection
def autocorrection(data):
    correct_spell = Speller(lang= 'en')
    correct_word = correct_spell(data)
    return correct_word

# 5. handling accented character
def accented_character(data):
    string_data = " ".join(data)
    fixed_text = unidecode(string_data)
    return fixed_text

# 6. lemmatization or stemming
def lemmatizer(data):
    lemma = WordNetLemmatizer()
    list1 = []
    for word in data.split():
#         print(word)
        lemmatized_word = lemma.lemmatize(word)
        list1.append(lemmatized_word)
    return " ".join(list1)
```

In [8]:
```
# what is data leakage ?
x_train, x_test, y_train, y_test = train_test_split(data.text, data.label, test_s
```

In [14]:
```
cleantext = remove_spaces(x_train[0])
print(cleantext)
cleantext = expand_text(cleantext)
print(cleantext)
cleantext = clean_text(cleantext,stopword_list)
print(cleantext)
cleantext = accented_character(cleantext)
print(cleantext)
cleantext = lemmatizer(cleantext)
print(cleantext)
```

```
I grew up (b. 1965) watching and loving the Thunderbirds. All my mates at sch
ool watched. We played "Thunderbirds" before school, during lunch and after s
chool. We all wanted to be Virgil or Scott. No one wanted to be Alan. Countin
g down from 5 became an art form. I took my children to see the movie hoping
they would get a glimpse of what I loved as a child. How bitterly disappointi
ng. The only high point was the snappy theme tune. Not that it could compare
with the original score of the Thunderbirds. Thankfully early Saturday mornin
gs one television channel still plays reruns of the series Gerry Anderson and
his wife created. Jonatha Frakes should hand in his directors chair, his vers
ion was completely hopeless. A waste of film. Utter rubbish. A CGI remake may
be acceptable but replacing marionettes with Homo sapiens subsp. sapiens was
a huge error of judgment.
I grew up (b. 1965) watching and loving the Thunderbirds. All my mates at sch
ool watched. We played "Thunderbirds" before school, during lunch and after s
chool. We all wanted to be Virgil or Scott. No one wanted to be Alan. Countin
g down from 5 became an art form. I took my children to see the movie hoping
they would get a glimpse of what I loved as a child. How bitterly disappointi
ng. The only high point was the snappy theme tune. Not that it could compare
with the original score of the Thunderbirds. Thankfully early Saturday mornin
gs one television channel still plays reruns of the series Gerry Anderson and
his wife created. Jonatha Frakes should hand in his directors chair, his vers
ion was completely hopeless. A waste of film. Utter rubbish. A CGI remake may
be acceptable but replacing marionettes with Homo sapiens subsp. sapiens was
a huge error of judgment.
['grew', 'watching', 'loving', 'thunderbirds', 'all', 'mates', 'school', 'wat
ched', 'played', 'thunderbirds', 'school', 'lunch', 'school', 'wanted', 'virg
il', 'scott', 'one', 'wanted', 'alan', 'counting', 'became', 'art', 'form',
'took', 'children', 'see', 'movie', 'hoping', 'would', 'get', 'glimpse', 'lov
ed', 'child', 'how', 'bitterly', 'disappointing', 'the', 'high', 'point', 'sn
appy', 'theme', 'tune', 'not', 'could', 'compare', 'original', 'score', 'thun
derbirds', 'thankfully', 'early', 'saturday', 'mornings', 'one', 'televisio
n', 'channel', 'still', 'plays', 'reruns', 'series', 'gerry', 'anderson', 'wi
fe', 'created', 'jonatha', 'frakes', 'hand', 'directors', 'chair', 'version',
'completely', 'hopeless', 'waste', 'film', 'utter', 'rubbish', 'cgi', 'remak
e', 'may', 'acceptable', 'replacing', 'marionettes', 'homo', 'sapiens', 'subs
p', 'sapiens', 'huge', 'error', 'judgment']
grew watching loving thunderbirds all mates school watched played thunderbird
s school lunch school wanted virgil scott one wanted alan counting became art
form took children see movie hoping would get glimpse loved child how bitterl
y disappointing the high point snappy theme tune not could compare original s
core thunderbirds thankfully early saturday mornings one television channel s
till plays reruns series gerry anderson wife created jonatha frakes hand dire
ctors chair version completely hopeless waste film utter rubbish cgi remake m
ay acceptable replacing marionettes homo sapiens subsp sapiens huge error jud
gment
```

```
grew watching loving thunderbird all mate school watched played thunderbird s
chool lunch school wanted virgil scott one wanted alan counting became art fo
rm took child see movie hoping would get glimpse loved child how bitterly dis
appointing the high point snappy theme tune not could compare original score
thunderbird thankfully early saturday morning one television channel still pl
ay rerun series gerry anderson wife created jonatha frakes hand director chai
r version completely hopeless waste film utter rubbish cgi remake may accepta
ble replacing marionette homo sapiens subsp sapiens huge error judgment
```

In [15]:
```
clean_text_train = x_train.apply(remove_spaces)
clean_text_test = x_test.apply(remove_spaces)

clean_text_train = clean_text_train.apply(expand_text)
clean_text_test = clean_text_test.apply(expand_text)


clean_text_train = clean_text_train.apply(lambda x :clean_text(x,stopword_list) )
clean_text_test = clean_text_test.apply(lambda x :clean_text(x,stopword_list))

clean_text_train = clean_text_train.apply(accented_character)
clean_text_test = clean_text_test.apply(accented_character)

clean_text_train = clean_text_train.apply(lemmatizer)
clean_text_test = clean_text_test.apply(lemmatizer)
```

In [16]:
```
clean_text_train
```

Out[16]:
```
12108    really thought would good movie boy mistaken f...
20671    kid movie great for family suck truly hoping s...
38174    really liked quirky movie the character not bl...
5388     rented creep not impressed not feel anything f...
11490    note not say better enjoyable the lack social ...
                               ...
29134    give movie worse cult movie deserve proper not...
16353    rififi chez le surface described french variat...
19674    good animation nice character design story mak...
7644     this movie plain bad not even worth watching m...
31689    forget movie forget many way outdated instead ...
Name: text, Length: 30000, dtype: object
```

1 Count Vectorizer

In [17]:
```
count_vect = CountVectorizer(max_features= 1000,max_df = 0.95,lowercase=True)
count_vect_train = count_vect.fit_transform(clean_text_train)
count_vect_test = count_vect.transform(clean_text_test)
```

In [18]:
```
count_vect_train.A # toarray(),.A
```

Out[18]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 1, 0, ..., 1, 0, 0]], dtype=int64)
```

In [19]:
```
pd.DataFrame(count_vect_train.A,columns=count_vect.get_feature_names())
```

Out[19]:

	ability	able	absolutely	accent	across	act	acted	acting	action	actor	...	wrote	yeah
0	0	0	0	0	0	0	0	2	0	1	...	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0
4	0	1	0	0	0	0	0	0	0	1	...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
29995	0	0	0	0	0	0	1	0	0	0	...	0	0
29996	0	0	0	0	0	0	0	2	0	2	...	0	0
29997	0	0	0	0	0	0	0	0	0	0	...	0	0
29998	0	0	0	0	0	0	0	0	0	0	...	0	0
29999	0	1	0	0	0	0	0	2	0	0	...	0	0

30000 rows × 1000 columns

In [20]:
```
count_mnb = MultinomialNB()
count_mnb.fit(count_vect_train.A,y_train)
```

Out[20]:
```
MultinomialNB
MultinomialNB()
```

In [21]:
```
pred_mnb_count = count_mnb.predict(count_vect_test.A)
pred_mnb_count
```

Out[21]:
```
array([0, 1, 0, ..., 0, 0, 0], dtype=int64)
```

In [22]:
```
accuracy_count = accuracy_score(y_test,pred_mnb_count)*100
accuracy_count
```

Out[22]:
```
83.28999999999999
```

2. Tfidf Vectorizer

In [23]:
```
tfidf_vect = TfidfVectorizer(max_features= 1000,max_df = 0.95,lowercase=True)
tfidf_vect_train = tfidf_vect.fit_transform(clean_text_train)
tfidf_vect_test = tfidf_vect.transform(clean_text_test)
pd.DataFrame(tfidf_vect_train.A,columns=tfidf_vect.get_feature_names())
```

Out[23]:

	ability	able	absolutely	accent	across	act	acted	acting	action	actor	...
0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.203035	0.0	0.104547	...
1	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.000000	...
2	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.000000	...
3	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.000000	...
4	0.0	0.104480	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.067656	...
...	...	...	...	...	...	...	...	...	...	...	...
29995	0.0	0.000000	0.0	0.0	0.0	0.0	0.167464	0.000000	0.0	0.000000	...
29996	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.054785	0.0	0.056420	...
29997	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.000000	...
29998	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.000000	...
29999	0.0	0.127929	0.0	0.0	0.0	0.0	0.000000	0.160880	0.0	0.000000	...

30000 rows × 1000 columns

In [24]:
```
tfidf_mnb = MultinomialNB()
tfidf_mnb.fit(tfidf_vect_train.A,y_train)
pred_mnb_tfidf = tfidf_mnb.predict(tfidf_vect_test.A)
accuracy_tfidf = accuracy_score(y_test,pred_mnb_tfidf)*100
accuracy_tfidf
```

Out[24]: 83.91999999999999

In [25]:
```
accuracy_count
```

Out[25]: 83.28999999999999

In [ ]:
```

```