In [1]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
```

In [2]:

```
data = pd.read_csv('IMDB Dataset.csv')
data.head()
```

Out[2]:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

In [3]:

```
data['sentiment'] = data['sentiment'].map({'positive':1, 'negative':2})
```

In [4]:

```
x=data['review']
y=data['sentiment']
```

In [5]:

```
x = [i.replace('<br />', '') for i in x]
x = [i.replace('\s', 's') for i in x]
```

In [62]:

```
y
```

Out[62]:

```
0        1
1        1
2        1
3        2
4        1
        ..
49995    1
49996    2
49997    2
49998    2
49999    2
Name: sentiment, Length: 50000, dtype: int64
```

In [63]:

```
vectorizer_tfidf = TfidfVectorizer(stop_words = 'english')
```

In [64]:

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.25,shuffle=False)
```

In [65]:

```
x_train_vec = vectorizer_tfidf.fit_transform(x_train)
```

In [66]:

```
print(x_train_vec)
```

```
  (0, 20533) 0.07594173231957477
  (0, 82825) 0.058367113261638195
  (0, 81267) 0.06383117990950907
  (0, 87467) 0.05355312015104057
  (0, 84913) 0.07299221723018025
  (0, 16917) 0.07407069790795343
  (0, 28405) 0.04897006964660852
  (0, 74273) 0.06644768319570975
  (0, 77914) 0.05620317654806377
  (0, 46008) 0.04950464905078183
  (0, 9455) 0.10052876834520749
  (0, 84185) 0.05062706876970766
  (0, 15902) 0.05263338925718544
  (0, 52363) 0.0509050167962195
  (0, 50087) 0.0857722821418323
  (0, 58094) 0.05078764838638907
  (0, 44874) 0.04903942808169445
  (0, 41317) 0.17228322419035963
  (0, 56065) 0.10625770612601676
  (0, 75397) 0.07064131950138874
  (0, 35583) 0.08093499100024586
  (0, 19483) 0.084564822718198
  (0, 41298) 0.08523938363879228
  (0, 34933) 0.06712275933726218
  (0, 47340) 0.0667980987382312
  : :
  (37499, 30147) 0.09622020278944149
  (37499, 6965) 0.1309297099469596
  (37499, 8499) 0.12270681417631772
  (37499, 54300) 0.09065970986361821
  (37499, 57549) 0.07096050632731758
  (37499, 37747) 0.23657400137732054
  (37499, 9531) 0.06778112325936907
  (37499, 79102) 0.060127429794995496
  (37499, 71851) 0.04547917899950881
  (37499, 2178) 0.09077085373209096
  (37499, 60464) 0.04282816348743424
  (37499, 41533) 0.06378896098707695
  (37499, 47615) 0.06323222042156293
  (37499, 49672) 0.08572584434608621
  (37499, 57594) 0.07669935202054814
  (37499, 54265) 0.07993042513262168
  (37499, 30026) 0.08862623668226324
  (37499, 81696) 0.05609385604337328
  (37499, 47530) 0.048387153291926346
  (37499, 50711) 0.09162618120515893
  (37499, 2218) 0.05235283532514523
  (37499, 72093) 0.06302517085098296
  (37499, 70703) 0.05734992651894421
  (37499, 49598) 0.19106805699830945
  (37499, 81567) 0.05182455010772026
```

In [67]:

```
x_test_vec = vectorizer_tfidf.transform(x_test)
print(x_test_vec)
```

```
  (0, 91602) 0.14939164825762885
  (0, 91223) 0.060471179399643166
  (0, 90878) 0.16066043393157478
  (0, 90744) 0.17573659136897052
  (0, 90375) 0.05916463986260935
  (0, 90173) 0.07402988543612228
  (0, 89604) 0.07072362455878622
  (0, 88755) 0.0845761577553766
```

```
  (0, 88419) 0.05779222340991027 6
  (0, 87233) 0.244785099600971
  (0, 86052) 0.1449081587245077
  (0, 83023) 0.10860533166756328
  (0, 80898) 0.12200114187744449
  (0, 79826) 0.14058343351783897
  (0, 79473) 0.09228750137816208
  (0, 78509) 0.09252400026602989
  (0, 77729) 0.08238879718460539
  (0, 74372) 0.16981615766103683
  (0, 73896) 0.07978359181439088
  (0, 73371) 0.11969700744282442
  (0, 73278) 0.1309467934322924
  (0, 72109) 0.11159721863638963
  (0, 70002) 0.06650811334915453
  (0, 68944) 0.06216874228691517
  (0, 67846) 0.22691577838921786
  : :
  (12499, 46974) 0.09794359428261766
  (12499, 45345) 0.1138746461510695
  (12499, 45059) 0.19096874142882783
  (12499, 43867) 0.05054150075646585
  (12499, 41708) 0.19323568942997374
  (12499, 40650) 0.1152344808347188
  (12499, 40343) 0.1741672401555403
  (12499, 37973) 0.09785428615621691
  (12499, 36681) 0.13151847110981862
  (12499, 34524) 0.16233389975795456
  (12499, 34461) 0.05310200906141328
  (12499, 29178) 0.17843815314426772
  (12499, 29134) 0.10991422758225962
  (12499, 29080) 0.10002987517058333
  (12499, 28377) 0.1891663665668354
  (12499, 28364) 0.11273415596515891
  (12499, 27279) 0.1272885941339425
  (12499, 19334) 0.20648381311692227
  (12499, 14552) 0.06996599731460251
  (12499, 14476) 0.17869714829829156
  (12499, 14427) 0.11480445518937489
  (12499, 12615) 0.15014810852393132
  (12499, 8906) 0.07151323000333204
  (12499, 6587) 0.12247433173871941
  (12499, 5573) 0.11499058269344122
```

In [68]:

```
nb = MultinomialNB()
nb.fit(x_train_vec, y_train)
```

Out[68]:

```
MultinomialNB()
```

In [69]:

```
nb.score(x_test_vec,y_test)
```

Out[69]:

```
0.86696
```

In [70]:

```
pred = nb.predict(x_test_vec)
```

In [71]:

```
print('Prediction Shape : ', pred.shape)
print('Actual Shape : ', y_test.shape)
```

```
Prediction Shape :  (12500,)
Actual Shape :  (12500,)
```

In [72]:

```
from sklearn.metrics import precision_score, recall_score, f1_score

print('Precision : ',precision_score(y_test, pred, average = "binary"))
print('Recall : ',recall_score(y_test, pred, average = "binary"))
print('F1 Score : ',f1_score(y_test, pred, average = "binary"))
```

```
Precision :  0.8784584980237155
Recall :  0.8522128135484902
F1 Score :  0.8651366474738464
```

In [73]:

```
my_review = ["From the co-writers of the Favorite, The Great is a terrific short series t
hat will leave you in stitches owing to its humor and sheer ridiculousness. It's the best
series that I've personally watched in a long time."]
my_review_neg = ["The more one sees the main characters, the less appealing they become.
Luke Skywalker is a whiner and just bad, Han Solo a bad sarcastic clod, Princess Leia a h
orrible nag, and C-3PO just a drone."]
my_review_pos = ["Dead To Me was a series that both of us binged on and we just couldn't
stop asking for more. We are so glad that this series received four Emmy nominations and
season three which would also be the final season (insert crying gif) is in the making!"]
```

In [74]:

```
vectorizer2 = TfidfVectorizer(stop_words = 'english')
my_review_vec = vectorizer2.fit_transform(my_review)
my_review_vec_pos = vectorizer_tfidf.transform(my_review_pos)
my_review_vec_neg = vectorizer_tfidf.transform(my_review_neg)
```

In [75]:

```
vectorizer2.vocabulary_
```

Out[75]:

```
{'writers': 17,
 'favorite': 1,
 'great': 2,
 'terrific': 13,
 'short': 11,
 'series': 9,
 'leave': 4,
 'stitches': 12,
 'owing': 6,
 'humor': 3,
 'sheer': 10,
 'ridiculousness': 8,
 'best': 0,
 've': 15,
 'personally': 7,
 'watched': 16,
 'long': 5,
 'time': 14}
```

In [76]:

```
pred_1 = nb.predict(my_review_vec_pos)
pred_1
```

Out[76]:

```
array([1])
```

In [77]:

```
pred_2 = nb.predict(my_review_vec_neg)
pred_2
```

Out[77]:

```
array([2])
```