```
['king', 'woman', 'she', 'lion', 'who', 'fox', 'brown', 'when', 'dare', 'cat']
```

.

SENTIMENT ANALYSIS ON AMAZON REVIEWS

In [34]:
```
import nltk
```

In [35]:
```
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\16193\AppData\Roaming\nltk_data...
```

Out[35]:
```
True
```

In [36]:
```
# Sentiment Intensity Analyzer takes in raw text and returns a dictionary of scores: ne

from nltk.sentiment.vader import SentimentIntensityAnalyzer
```

In [37]:
```
sid = SentimentIntensityAnalyzer()
```

In [38]:
```
# Use it for example strings

a = "This is a good movie"
```

In [40]:
```
sid.polarity_scores(a)
```

Out[40]:
```
{'neg': 0.0, 'neu': 0.508, 'pos': 0.492, 'compound': 0.4404}
```

In [41]:
```
a = "This was the best, most awesome movie EVER MADE!!!"
```

In [42]:
```
sid.polarity_scores(a)
```

Out[42]:
```
{'neg': 0.0, 'neu': 0.425, 'pos': 0.575, 'compound': 0.8877}
```

In [43]:
```
a = "This was the WORST movie that has ever disgraced the screen."
```

In [44]:
```
sid.polarity_scores(a)
```

Out[44]:
```
{'neg': 0.465, 'neu': 0.535, 'pos': 0.0, 'compound': -0.8331}
```

In [48]:
```
# USE VADER FOR AMAZON REVIEWS

import pandas as pd

df = pd.read_csv('amazonreviews.tsv',sep='\t')


df.head()
```

Out[48]:

	label	review
0	pos	Stuning even for the non-gamer: This sound tra...
1	pos	The best soundtrack ever to anything.: I'm rea...
2	pos	Amazing!: This soundtrack is my favorite music...

	label	review
3	pos	Excellent Soundtrack: I truly like this soundt...
4	pos	Remember, Pull Your Jaw Off The Floor After He...

In [49]:
```
df['label'].value_counts()
```

Out[49]:
```
neg    5097
pos    4903
Name: label, dtype: int64
```

In [50]:
```
# Clean data (drop null values first)

df.dropna(inplace=True)
```

In [51]:
```
# Clean data (remove all reviews with blanks)

blanks = []

for i, lb, rv in df.itertuples():
    # (index, label, review)
    if type(rv) == str:
        if rv.isspace():
            blanks.append(i)
```

In [52]:
```
# Good news: no blanks

blanks
```

Out[52]:
```
[]
```

In [53]:
```
df.drop(blanks,inplace=True)
```

In [54]:
```
df.iloc[0]['review']
```

Out[54]:
```
'Stuning even for the non-gamer: This sound track was beautiful! It paints the senery in your mind so well I would recomend it even to people who hate vid. game music! I have played the game Chrono Cross but out of all of the games I have ever played it has the best music! It backs away from crude keyboarding and takes a fresher step with grate guitars and soulful orchestras. It would impress anyone who cares to listen! ^_^'
```

In [55]:
```
# What are the scores for this Amazon review?

sid.polarity_scores(df.iloc[0]['review'])
```

Out[55]:
```
{'neg': 0.088, 'neu': 0.669, 'pos': 0.243, 'compound': 0.9454}
```

In [56]:
```
# To apply VADER sentiment scores to every review

df['scores'] = df['review'].apply(lambda review: sid.polarity_scores(review))
```

In [57]:
```
df.head()
```

Out[57]:

	label	review	scores
0	pos	Stuning even for the non-gamer: This sound tra...	{'neg': 0.088, 'neu': 0.669, 'pos': 0.243, 'co...

	label	review	scores
1	pos	The best soundtrack ever to anything.: I'm rea...	{'neg': 0.018, 'neu': 0.837, 'pos': 0.145, 'co...
2	pos	Amazing!: This soundtrack is my favorite music...	{'neg': 0.04, 'neu': 0.692, 'pos': 0.268, 'com...
3	pos	Excellent Soundtrack: I truly like this soundt...	{'neg': 0.09, 'neu': 0.615, 'pos': 0.295, 'com...
4	pos	Remember, Pull Your Jaw Off The Floor After He...	{'neg': 0.0, 'neu': 0.746, 'pos': 0.254, 'comp...

In [58]:
```
df['compound'] = df['scores'].apply(lambda d:d['compound'])
```

In [59]:
```
# Create a column that has only compound scores

df.head()
```

Out[59]:

	label	review	scores	compound
0	pos	Stuning even for the non-gamer: This sound tra...	{'neg': 0.088, 'neu': 0.669, 'pos': 0.243, 'co...	0.9454
1	pos	The best soundtrack ever to anything.: I'm rea...	{'neg': 0.018, 'neu': 0.837, 'pos': 0.145, 'co...	0.8957
2	pos	Amazing!: This soundtrack is my favorite music...	{'neg': 0.04, 'neu': 0.692, 'pos': 0.268, 'com...	0.9858
3	pos	Excellent Soundtrack: I truly like this soundt...	{'neg': 0.09, 'neu': 0.615, 'pos': 0.295, 'com...	0.9814
4	pos	Remember, Pull Your Jaw Off The Floor After He...	{'neg': 0.0, 'neu': 0.746, 'pos': 0.254, 'comp...	0.9781

In [60]:
```
# If greater than 0, positive
# If less than 0, negative

df['comp_score'] = df['compound'].apply(lambda score: 'pos' if score >=0 else 'neg')
```

In [63]:
```
df.head(11)
```

Out[63]:

	label	review	scores	compound	comp_score
0	pos	Stuning even for the non-gamer: This sound tra...	{'neg': 0.088, 'neu': 0.669, 'pos': 0.243, 'co...	0.9454	pos
1	pos	The best soundtrack ever to anything.: I'm rea...	{'neg': 0.018, 'neu': 0.837, 'pos': 0.145, 'co...	0.8957	pos
2	pos	Amazing!: This soundtrack is my favorite music...	{'neg': 0.04, 'neu': 0.692, 'pos': 0.268, 'com...	0.9858	pos
3	pos	Excellent Soundtrack: I truly like this soundt...	{'neg': 0.09, 'neu': 0.615, 'pos': 0.295, 'com...	0.9814	pos
4	pos	Remember, Pull Your Jaw Off The Floor After He...	{'neg': 0.0, 'neu': 0.746, 'pos': 0.254, 'comp...	0.9781	pos
5	pos	an absolute masterpiece: I am quite sure any o...	{'neg': 0.014, 'neu': 0.737, 'pos': 0.249, 'co...	0.9900	pos

	label	review	scores	compound	comp_score
6	neg	Buyer beware: This is a self-published book, a...	{'neg': 0.124, 'neu': 0.806, 'pos': 0.069, 'co...	-0.8744	neg
7	pos	Glorious story: I loved Whisper of the wicked ...	{'neg': 0.072, 'neu': 0.583, 'pos': 0.346, 'co...	0.9900	pos
8	pos	A FIVE STAR BOOK: I just finished reading Whis...	{'neg': 0.113, 'neu': 0.712, 'pos': 0.174, 'co...	0.8353	pos
9	pos	Whispers of the Wicked Saints: This was a easy...	{'neg': 0.033, 'neu': 0.777, 'pos': 0.19, 'com...	0.8196	pos
10	neg	The Worst!: A complete waste of time. Typograp...	{'neg': 0.36, 'neu': 0.586, 'pos': 0.054, 'com...	-0.9274	neg

In [64]:
```
# How accurate is the VADER score to the already-named label score?

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

In [65]:
```
accuracy_score(df['label'],df['comp_score'])
```

Out[65]:
```
0.7091
```

In [66]:
```
print(classification_report(df['label'],df['comp_score']))
```

```
              precision    recall  f1-score   support

         neg       0.86      0.51      0.64      5097
         pos       0.64      0.91      0.75      4903

    accuracy                           0.71     10000
   macro avg       0.75      0.71      0.70     10000
weighted avg       0.75      0.71      0.70     10000
```

In [67]:
```
# VADER has some trouble with negative reviews, some are hard to read and some are sarc
```

In [68]:
```
print(confusion_matrix(df['label'],df['comp_score']))
```

```
[[2622 2475]
 [ 434 4469]]
```

In [69]:
```
# VADER was accurate to 71% of the data. Deep Learning sentiment analysis will have hig
```

.

SENTIMENT ANALYSIS ON MOVIE REVIEWS

In [70]:
```
import numpy as np
import pandas as pd
```

In [71]:
```
df = pd.read_csv('moviereviews.tsv',sep='\t')
```

In [72]:
```
df.head()
```

Out[72]:

	label	review

	label	review
0	neg	how do films like mouse hunt get into theatres...
1	neg	some talented actresses are blessed with a dem...
2	pos	this has been an extraordinary year for austra...
3	pos	according to hollywood movies made in last few...
4	neg	my first press screening of 1998 and already i...

In [74]:
```
df.dropna(inplace=True)
```

In [75]:
```
blanks = []

for i, lb, rv in df.itertuples():
    if type(rv) == str:
        if rv.isspace():
            blanks.append(i)
```

In [76]:
```
blanks
```

Out[76]:
```
[57,
 71,
 147,
 151,
 283,
 307,
 313,
 323,
 343,
 351,
 427,
 501,
 633,
 675,
 815,
 851,
 977,
 1079,
 1299,
 1455,
 1493,
 1525,
 1531,
 1763,
 1851,
 1905,
 1993]
```

In [77]:
```
df.drop(blanks,inplace=True)
```

In [78]:
```
df['label'].value_counts()
```

Out[78]:
```
pos    969
neg    969
Name: label, dtype: int64
```

In [79]:
```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
```

In [80]:
```
sid = SentimentIntensityAnalyzer()
```

In [81]:
```
df['scores'] = df['review'].apply(lambda review: sid.polarity_scores(review))
```

In [82]:
```
df['compound'] = df['scores'].apply(lambda d: d['compound'])
```

In [83]:
```
df.head()
```

Out[83]:

	label	review	scores	compound
0	neg	how do films like mouse hunt get into theatres...	{'neg': 0.121, 'neu': 0.778, 'pos': 0.101, 'co...	-0.9125
1	neg	some talented actresses are blessed with a dem...	{'neg': 0.12, 'neu': 0.775, 'pos': 0.105, 'com...	-0.8618
2	pos	this has been an extraordinary year for austra...	{'neg': 0.068, 'neu': 0.781, 'pos': 0.15, 'com...	0.9951
3	pos	according to hollywood movies made in last few...	{'neg': 0.071, 'neu': 0.782, 'pos': 0.147, 'co...	0.9972
4	neg	my first press screening of 1998 and already i...	{'neg': 0.091, 'neu': 0.817, 'pos': 0.093, 'co...	-0.2484

In [84]:
```
df['comp_score'] = df['compound'].apply(lambda score: 'pos' if score >= 0 else 'neg')
```

In [86]:
```
df.head(11)
```

Out[86]:

	label	review	scores	compound	comp_score
0	neg	how do films like mouse hunt get into theatres...	{'neg': 0.121, 'neu': 0.778, 'pos': 0.101, 'co...	-0.9125	neg
1	neg	some talented actresses are blessed with a dem...	{'neg': 0.12, 'neu': 0.775, 'pos': 0.105, 'com...	-0.8618	neg
2	pos	this has been an extraordinary year for austra...	{'neg': 0.068, 'neu': 0.781, 'pos': 0.15, 'com...	0.9951	pos
3	pos	according to hollywood movies made in last few...	{'neg': 0.071, 'neu': 0.782, 'pos': 0.147, 'co...	0.9972	pos
4	neg	my first press screening of 1998 and already i...	{'neg': 0.091, 'neu': 0.817, 'pos': 0.093, 'co...	-0.2484	neg
5	neg	to put it bluntly , ed wood would have been pr...	{'neg': 0.123, 'neu': 0.821, 'pos': 0.056, 'co...	-0.9855	neg
6	neg	synopsis : melissa , a mentally-disturbed woma...	{'neg': 0.087, 'neu': 0.742, 'pos': 0.17, 'com...	0.9871	pos
7	neg	tim robbins and martin lawernce team up in thi...	{'neg': 0.118, 'neu': 0.709, 'pos': 0.172, 'co...	0.9829	pos
8	neg	in " gia " , angelina jolie plays the titular ...	{'neg': 0.082, 'neu': 0.862, 'pos': 0.056, 'co...	-0.8278	neg
9	neg	in 1990 , the surprise success an unheralded l...	{'neg': 0.145, 'neu': 0.728, 'pos': 0.127, 'co...	-0.9147	neg

	label	review	scores	compound	comp_score
10	neg	upon first viewing of this movie , the phrases...	{'neg': 0.114, 'neu': 0.742, 'pos': 0.143, 'co...	0.9544	pos

In [87]:
```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

In [88]:
```
accuracy_score(df['label'],df['comp_score'])
```

Out[88]:
```
0.6357069143446853
```

In [89]:
```
print(classification_report(df['label'],df['comp_score']))
```

```
              precision    recall  f1-score   support

         neg       0.72      0.44      0.55       969
         pos       0.60      0.83      0.70       969

    accuracy                           0.64      1938
   macro avg       0.66      0.64      0.62      1938
weighted avg       0.66      0.64      0.62      1938
```

In [90]:
```
print(confusion_matrix(df['label'],df['comp_score']))
```

```
[[427 542]
 [164 805]]
```

In [91]:
```
# Again, sarcasm reviews make it really hard to detect if a review is positive or negat

# One of the biggest challenges in Sentiment Analysis is a computer understanding human
```

In [ ]:
```

```