In [1]:
```
import nltk
```

In [2]:
```
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     /Users/jayashrijagannathan/nltk_data...
```

Out[2]:
```
True
```

In [3]:
```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
```

In [5]:
```
sid = SentimentIntensityAnalyzer()
```

In [7]:
```
a = "This is a good movie"
sid.polarity_scores(a)
```

Out[7]:
```
{'neg': 0.0, 'neu': 0.508, 'pos': 0.492, 'compound': 0.4404}
```

In [10]:
```
a = "This was the best, most awesome movie EVER MADE!!!"
sid.polarity_scores(a)
```

Out[10]:
```
{'neg': 0.0, 'neu': 0.425, 'pos': 0.575, 'compound': 0.8877}
```

In [11]:
```
a = "This was the WORST movie that ever disgraced the screen."
```

In [12]:
```
sid.polarity_scores(a)
```

Out[12]:
```
{'neg': 0.495, 'neu': 0.505, 'pos': 0.0, 'compound': -0.8331}
```

Use Vader to analyze Amazon Reviews

In [13]:
```
import pandas as pd
import numpy as np
```

In [14]:
```
df = pd.read_csv('../TextFiles/amazonreviews.tsv', sep = '\t')
```

In [15]:
```
df.head()
```

Out[15]:

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

In [16]:
```
df.isnull().sum()
```

Out[16]:
```
label     0
review    0
dtype: int64
```

In [17]:
```
df['label'].value_counts()
```

Out[17]:
```
neg    5097
pos    4903
Name: label, dtype: int64
```

In [18]:
```
# Check for blanks
blanks = []

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

print(blanks)
```

```
[]
```

In [19]:
```
# If we had null rows-- drop them
# df.dropna(inplace = True

# If we had blanks -- then drop the rows with the index
#df.drop(blanks, inplace = True)
```

Analyze first Amazon Review using VADER

In [22]:
```
sid.polarity_scores(df.iloc[0]['review'])
```

Out[22]:
```
{'neg': 0.088, 'neu': 0.669, 'pos': 0.243, 'compound': 0.9454}
```

In [21]:
```
# Compare with the first review with the label
df.iloc[0]['label']
```

Out[21]:
```
'pos'
```

In [23]:
```
df.iloc[0]['review']
```

Out[23]:
```
'Stuning even for the non-gamer: This sound track was beautiful! It pa
ints the senery in your mind so well I would recomend it even to peopl
e who hate vid. game music! I have played the game Chrono Cross but ou
t of all of the games I have ever played it has the best music! It bac
ks away from crude keyboarding and takes a fresher step with grate gui
tars and soulful orchestras. It would impress anyone who cares to list
en! ^_^'
```

In [24]:
```
# Create a new column 'scores' of the polarity scores
df['scores'] = df['review'].apply(lambda x: sid.polarity_scores(x))
```

In [25]:
```
df.head()
```

Out[25]:

	label	review	scores
0	pos	Stuning even for the non-gamer: This sound tra...	{'neg': 0.088, 'neu': 0.669, 'pos': 0.243, 'co...
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 [27]:
```
df['scores'].head()
```

Out[27]:
```
0    {'neg': 0.088, 'neu': 0.669, 'pos': 0.243, 'co...
1    {'neg': 0.018, 'neu': 0.837, 'pos': 0.145, 'co...
2    {'neg': 0.04, 'neu': 0.692, 'pos': 0.268, 'com...
3    {'neg': 0.09, 'neu': 0.615, 'pos': 0.295, 'com...
4    {'neg': 0.0, 'neu': 0.746, 'pos': 0.254, 'comp...
Name: scores, dtype: object
```

In [28]:
```
# Create a new column called 'compound' having compound polarity score
df['compound'] = df['scores'].apply(lambda d: d['compound'])
```

In [29]:
```
df.head()
```

Out[29]:

	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 [30]:
```
# Create a new columnn 'comp_score' which is 'pos' if compound >= 0
# and 'neg' if compound < 0

df['comp_score'] = df['compound'].apply(lambda x:'pos' if x >= 0 else 'n
df.head()
```

Out[30]:

	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

Report Metrics on Accuracy of VADER

In [34]:
```
from sklearn.metrics import confusion_matrix, classification_report, acc
```

In [36]:
```
print(accuracy_score(df['label'], df['comp_score']))
```

```
0.7091
```

In [37]:
```
print(confusion_matrix(df['label'], df['comp_score']))
```

```
[[2623 2474]
 [ 435 4468]]
```

In [39]:
```
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

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

In [40]:
```
# Create a column called label_val. It is 1 if label= 'pos' and 0 if 'ne
df['label_val'] = df['label'].apply(lambda x: 1 if x == 'pos' else 0)
```

In [43]:
```
# Create a column called comp_val. It is 1 if comp_score= 'pos' and 0 if
df['comp_val'] = df['comp_score'].apply(lambda x: 1 if x == 'pos' else 0
```

In [45]:
```
df[['label_val', 'comp_val']].head()
```

Out[45]:

	label_val	comp_val
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1

In [46]:
```
df[['label_val', 'comp_val']].corr()
```

Out[46]:

	label_val	comp_val
label_val	1.000000	0.462094
comp_val	0.462094	1.000000

In [48]:
```
# Correlation between label_val and compound
df[['label_val', 'compound']].head()
```

Out[48]:

	label_val	compound
0	1	0.9454
1	1	0.8957
2	1	0.9858
3	1	0.9814
4	1	0.9781

In [49]:
```
df[['label_val', 'compound']].corr()
```

Out[49]:

	label_val	compound
label_val	1.000000	0.531163
compound	0.531163	1.000000

In [ ]: