

Download vader_lexicon (Used for Sentiment Analysis)

In []:

```
import nltk
nltk.download('vader_lexicon')
```

Import Libraries

In [7]:

```
import numpy as np
import pandas as pd
```

Dataset

In [9]:

```
df = pd.read_csv('Restaurant_Reviews.tsv', sep='\t')
df.head()
```

Out[9]:

	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday of...	1
4	The selection on the menu was great and so wer...	1

Liked Column Counts

In [10]:

```
df['Liked'].value_counts()
```

Out[10]:

```
0    500
1    500
Name: Liked, dtype: int64
```

Text Cleaning

In [11]:

```
df.dropna(inplace=True) #removes null values

blanks=[]

for i,lb,rw in df.iterrows():

    if type(rw)==str:

        if rw.isspace(): #avoid NAN values

            blanks.append(i) #add matching index numbers to the list

df.drop(blanks, inplace=True)
```

Checking count after cleaning text

In [12]:

```
df['Liked'].value_counts()
```

Out[12]:

```
0    500
1    500
Name: Liked, dtype: int64
```

Sentiment Intensity Analyzer

In [13]:

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sid=SentimentIntensityAnalyzer()
```

Polarity Classification Score

In [14]:

```
sid.polarity_scores(df.loc[0]['Review'])
```

Out[14]:

```
{'neg': 0.0, 'neu': 0.435, 'pos': 0.565, 'compound': 0.5994}
```

In [15]:

```
df.loc[0]['Liked']
```

Out[15]:

```
1
```

Adding Polarity Scores to DataFrame

In [17]:

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

Out[17]:

	Review	Liked	scores
0	Wow... Loved this place.	1	{'neg': 0.0, 'neu': 0.435, 'pos': 0.565, 'comp...
1	Crust is not good.	0	{'neg': 0.445, 'neu': 0.555, 'pos': 0.0, 'comp...
2	Not tasty and the texture was just nasty.	0	{'neg': 0.34, 'neu': 0.66, 'pos': 0.0, 'compou...
3	Stopped by during the late May bank holiday of...	1	{'neg': 0.093, 'neu': 0.585, 'pos': 0.322, 'co...
4	The selection on the menu was great and so wer...	1	{'neg': 0.0, 'neu': 0.728, 'pos': 0.272, 'comp...

Adding Compound Value to DataFrame

In [18]:

```
df['compound'] = df['scores'].apply(lambda score_dict: score_dict['compound'])
df.head()
```

Out[18]:

	Review	Liked	scores	compound
0	Wow... Loved this place.	1	{'neg': 0.0, 'neu': 0.435, 'pos': 0.565, 'comp...	0.5994
1	Crust is not good.	0	{'neg': 0.445, 'neu': 0.555, 'pos': 0.0, 'comp...	-0.3412
2	Not tasty and the texture was just nasty.	0	{'neg': 0.34, 'neu': 0.66, 'pos': 0.0, 'compou...	-0.5574
3	Stopped by during the late May bank holiday of...	1	{'neg': 0.093, 'neu': 0.585, 'pos': 0.322, 'co...	0.6908
4	The selection on the menu was great and so wer...	1	{'neg': 0.0, 'neu': 0.728, 'pos': 0.272, 'comp...	0.6249

Adding Compound Score to DataFrame

In [19]:

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

Out[19]:

	Review	Liked	scores	compound	comp_score
0	Wow... Loved this place.	1	{'neg': 0.0, 'neu': 0.435, 'pos': 0.565, 'comp...	0.5994	pos
1	Crust is not good.	0	{'neg': 0.445, 'neu': 0.555, 'pos': 0.0, 'comp...	-0.3412	neg
2	Not tasty and the texture was just nasty.	0	{'neg': 0.34, 'neu': 0.66, 'pos': 0.0, 'compou...	-0.5574	neg
3	Stopped by during the late May bank holiday of...	1	{'neg': 0.093, 'neu': 0.585, 'pos': 0.322, 'co...	0.6908	pos
4	The selection on the menu was great and so wer...	1	{'neg': 0.0, 'neu': 0.728, 'pos': 0.272, 'comp...	0.6249	pos

Converting Compound Score to Integer

In [23]:

```
from sklearn.preprocessing import LabelEncoder
```

In [25]:

```
lb=LabelEncoder()
```

In [26]:

```
df['comp_score'] = lb.fit_transform(df['comp_score'])
df.head()
```

Out[26]:

	Review	Liked	scores	compound	comp_score
0	Wow... Loved this place.	1	{'neg': 0.0, 'neu': 0.435, 'pos': 0.565, 'comp...	0.5994	1
1	Crust is not good.	0	{'neg': 0.445, 'neu': 0.555, 'pos': 0.0, 'comp...	-0.3412	0
2	Not tasty and the texture was just nasty.	0	{'neg': 0.34, 'neu': 0.66, 'pos': 0.0, 'compou...	-0.5574	0
3	Stopped by during the late May bank holiday of...	1	{'neg': 0.093, 'neu': 0.585, 'pos': 0.322, 'co...	0.6908	1
4	The selection on the menu was great and so wer...	1	{'neg': 0.0, 'neu': 0.728, 'pos': 0.272, 'comp...	0.6249	1

Predictions and Accuracy

In [27]:

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

In [28]:

```
accuracy_score(df['Liked'], df['comp_score'])
```

Out[28]:

0.723

Accuracy : 72%

In [29]:

```
print(classification_report(df['Liked'], df['comp_score']))
```

	precision	recall	f1-score	support
0	0.93	0.48	0.63	500
1	0.65	0.97	0.78	500
accuracy			0.72	1000
macro avg	0.79	0.72	0.71	1000
weighted avg	0.79	0.72	0.71	1000

Confusion Matrix

In [30]:

```
print(confusion_matrix(df['Liked'], df['comp_score']))
```

```
[[240 260]
 [ 17 483]]
```

In []: