

In [3]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [4]:

```
df = pd.read_csv('/content/Restaurant_Reviews.tsv', delimiter='\t', quoting=3)
df.head()
```

Out[4]:

| | 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 |

In [5]:

```
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []

for i in range(0,1000):
    zomato_ratings = re.sub('[^a-zA-Z]', ' ', df['Review'][i])
    zomato_ratings = zomato_ratings.lower()
    zomato_ratings = zomato_ratings.split()
    ps = PorterStemmer()
    all_stop = stopwords.words('english')
    all_stop.remove('not')
    zomato_ratings = [ps.stem(word) for word in zomato_ratings if not word in set(all_stop)]
    zomato_ratings = ' '.join(zomato_ratings)
    corpus.append(zomato_ratings)
```

[nltk_data] Downloading package stopwords to /root/nltk_data...

[nltk_data] Unzipping corpora/stopwords.zip.

In [43]:

```
print(corpus[0])
print(corpus[1])
print(corpus[2])
```

```
wow love place
crust not good
not tasti textur nasti
```

In [36]:

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
X = cv.fit_transform(corpus).toarray()
y = df.iloc[:, -1].values
# print(cv.vocabulary_)
```

In [32]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state
= 21)
```

In [37]:

```
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(X_train, y_train)
```

Out[37]:

```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

In [38]:

```
y_pred = classifier.predict(X_test)
a = np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1)
a[5]
```

Out[38]:

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

In [39]:

```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
classifier.score(X_test, y_test)
```

```
[[73 29]
 [20 78]]
```

Out[39]:

```
0.755
```

In [40]:

```
c=0
for i in y_pred:
    if c!=5:
        if(i==1):
            print("Positive Review")
            c = c+1
        else:
            print("Negative Review")
            c = c+1
```

```
Negative Review
Negative Review
Positive Review
Positive Review
Positive Review
```