In [1]:

```
import pandas as pd
import numpy as np
import nltk
import re
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
```

In [3]:

```
data = pd.read_csv("a1_RestaurantReviews_HistoricDump.tsv" , delimiter="\t")
```

In [4]:

```
data.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]:

```
data.shape
```

Out[5]:

```
(900, 2)
```

Data Cleaning

In [6]:

```
ps = PorterStemmer()
stop_words = stopwords.words("english")
stop_words.remove("not")
```

In [7]:

```
corpus = []
for i in range(0 , 900):
    text = re.sub("[^a-zA-Z]" , " " , data["Review"][i])
    text = text.lower()
    text = text.split()
    text = [ps.stem(word) for word in text if not word in set(stop_words)]
    text = " ".join(text)
    corpus.append(text)
```

In [8]:

```
corpus
```

Out[8]:

```
['wow love place',
 'crust not good',
 'not tasti textur nasti',
 'stop late may bank holiday rick steve recommend love',
 'select menu great price',
 'get angri want damn pho',
 'honeslti tast fresh',
 'potato like rubber could tell made ahead time kept warmer',
 'fri great',
 'great touch',
 'servic prompt',
 'would not go back',
 'cashier care ever say still end wayyy overpr',
 'tri cape cod ravoli chicken cranberri mmmm',
 'disgust pretti sure human hair',
 'shock sign indic cash',
 'highli recommend',
 'waitress littl slow servic',
```

Data Transformation

In [9]:

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=1420)
```

In [10]:

```
x = cv.fit_transform(corpus).toarray()
y = data["Liked"]
```

In [11]:

```
# saving the bow for later use in prediction
import pickle
bow_path = "./bow_model.pkl"
pickle.dump(cv , open(bow_path , "wb"))
```

In [12]:

```
from sklearn.model_selection import train_test_split
```

In [13]:

```
x_train , x_test , y_train, y_test = train_test_split(x, y , test_size=0.2 , random_s
```

Building Model

In [14]:

```
from sklearn.naive_bayes import GaussianNB
classifer = GaussianNB()
classifer.fit(x_train , y_train)
```

Out[14]:

```
GaussianNB()
```

In [15]:

```
# saving the model for later use in prediction
model_path = "./model.pkl"
pickle.dump(classifer , open(model_path , "wb"))
```

In [16]:

```
y_pred = classifer.predict(x_test)
```

In [17]:

```
from sklearn.metrics import confusion_matrix , accuracy_score

cm = confusion_matrix(y_pred , y_test)
```

In [18]:

```
cm
```

Out[18]:

```
array([[47, 14],
       [35, 84]], dtype=int64)
```

In [19]:

```
score = accuracy_score(y_test , y_pred)
```

In [20]:

```
score
```

Out[20]:

```
0.7277777777777777
```

Prediction using Fresh Dataset

In [21]:

```
df = pd.read_csv("a2_RestaurantReviews_FreshDump.tsv" , delimiter="\t")
```

In [22]:

```
df.head()
```

Out[22]:

	Review
0	Spend your money elsewhere.
1	Their regular toasted bread was equally satisf...
2	The Buffet at Bellagio was far from what I ant...
3	And the drinks are WEAK, people!
4	-My order was not correct.

In [23]:

```
df.shape
```

Out[23]:

```
(100, 1)
```

Data Cleaning

In [24]:

```
ps = PorterStemmer()
stop_words = stopwords.words("english")
stop_words.remove("not")
```

In [25]:

```
corpus = []
for i in range(0 , 100):
    text = re.sub("[^a-zA-Z]" , " ", df["Review"][i])
    text = text.lower()
    text = text.split()
    text = [ps.stem(word) for word in text if not word in set(stop_words)]
    text = " ".join(text)
    corpus.append(text)
```

data Transformation

In [26]:

```
# loading the bow file
from sklearn.feature_extraction.text import CountVectorizer
import pickle
cv_path = "./bow_model.pkl"
cv = pickle.load(open(cv_path ,"rb"))
```

In [27]:

```
x_fresh = cv.transform(corpus).toarray()
x_fresh.shape
```

Out[27]:

```
(100, 1420)
```

Model Building

In [28]:

```
# loading classifer model
classifer = pickle.load(open("./model.pkl" , "rb"))
```

In [29]:

```
y_pred = classifer.predict(x_fresh)
```

In [30]:

```
y_pred
```

Out[30]:

```
array([0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0], dtype=int64)
```

In [31]:

```
df["Predicted Labels"] = y_pred.tolist()
```

In [32]:

```
df.sample(10)
```

Out[32]:

	Review	Predicted Labels
41	Probably not in a hurry to go back.	0
48	This place is horrible and way overpriced.	0
38	The meat was pretty dry, I had the sliced bris...	0
64	Del Taco is pretty nasty and should be avoided...	0
88	It really is impressive that the place hasn't ...	0
51	The tables outside are also dirty a lot of the...	0
31	If you want to wait for mediocre food and down...	0
97	Overall I was not impressed and would not go b...	0
9	This is my new fav Vegas buffet spot.	1
25	I could barely stomach the meal, but didn't co...	0

In [32]:

```
df.to_csv("./Predicted Sentiments Fresh Data.csv", sep="\t")
```

In [ ]:

```

```