```
import pandas as pd
```

```
df=pd.read_csv("/content/zomato.csv")
```

```
df
```

	Unnamed: 0.1	Unnamed: 0	restaurant name	restaurant type	rate (out of 5)	num of ratings	avg cost (two people)	online_order	table booking	cuisines type	
0	0	0	#FeelTheROLL	Quick Bites	3.4	7	200.0	No	No	Fast Food	Be
1	1	1	#L-81 Cafe	Quick Bites	3.9	48	400.0	Yes	No	Fast Food, Beverages	Byresandra,Tavarekere,Ma
2	2	2	#refuel	Cafe	3.7	37	400.0	Yes	No	Cafe, Beverages	Bannerghatt
3	3	3	'@ Biryani Central	Casual Dining	2.7	135	550.0	Yes	No	Biryani, Mughlai, Chinese	Mara
4	4	4	'@ The Bbq	Casual Dining	2.8	40	700.0	Yes	No	BBQ, Continental, North Indian, Chinese, Bever...	Be
...	...	...	...	...	...	...	...	...	...	...	
7100	7100	7100	Zoey's	Cafe	4.3	894	600.0	Yes	No	Cafe, Italian, Continental, Burger	Be
7101	7101	7101	ZOROY Luxury Chocolate	Dessert Parlor	4.0	68	250.0	Yes	No	Desserts	Brigad
7102	7102	7102	Zu's Doner Kebaps	Takeaway, Delivery	3.7	33	350.0	No	No	Turkish, Fast Food, Biryani, Chinese	Mallesh
7103	7103	7103	Zyara	Casual Dining	3.8	191	650.0	Yes	No	North Indian, Mughlai, Chinese	Kamma

```
df.columns
```

```
Index(['Unnamed: 0.1', 'Unnamed: 0', 'restaurant name', 'restaurant type',
       'rate (out of 5)', 'num of ratings', 'avg cost (two people)',
       'online_order', 'table booking', 'cuisines type', 'area',
       'local address'],
      dtype='object')
```

1

```
df.isnull().sum()
```

```
Unnamed: 0.1              0
Unnamed: 0                0
restaurant name           0
restaurant type           0
rate (out of 5)          68
num of ratings            0
avg cost (two people)    57
online_order              0
table booking             0
cuisines type             0
area                      0
local address             0
dtype: int64
```

```
df.drop(columns=['Unnamed: 0.1', 'Unnamed: 0'], inplace=True)
```

```
df
```

	restaurant name	restaurant type	rate (out of 5)	num of ratings	avg cost (two people)	online_order	table booking	cuisines type	area	local address
0	#FeelTheROLL	Quick Bites	3.4	7	200.0	No	No	Fast Food	Bellandur	Bellandur
1	#L-81 Cafe	Quick Bites	3.9	48	400.0	Yes	No	Fast Food, Beverages	Byresandra,Tavarekere,Madiwala	HSR
2	#refuel	Cafe	3.7	37	400.0	Yes	No	Cafe, Beverages	Bannerghatta Road	Bannerghatta Road
3	'@ Biryani Central	Casual Dining	2.7	135	550.0	Yes	No	Biryani, Mughlai, Chinese	Marathahalli	Marathahalli
4	'@ The Bbq	Casual Dining	2.8	40	700.0	Yes	No	BBQ, Continental, North Indian, Chinese, Bever...	Bellandur	Bellandur
...	...	...	...	...	...	...	...	...	...	...
7100	Zoey's	Cafe	4.3	894	600.0	Yes	No	Cafe, Italian, Continental, Burger	Bellandur	Sarjapur Road
7101	ZOROY Luxury Chocolate	Dessert Parlor	4.0	68	250.0	Yes	No	Desserts	Brigade Road	Church Street
7102	Zu's Doner	Takeaway,	3.7	33	350.0	No	No	Turkish, Fast Food,	Malleshwaram	RT Nagar

```
df['popularity'] = df['rate (out of 5)'].apply(lambda x: 'High' if x > 4 else 'Low')

df.head()
```

	restaurant name	restaurant type	rate (out of 5)	num of ratings	avg cost (two people)	online_order	table booking	cuisines type	area	local address	po
0	#FeelTheROLL	Quick Bites	3.4	7	200.0	No	No	Fast Food	Bellandur	Bellandur	
1	#L-81 Cafe	Quick Bites	3.9	48	400.0	Yes	No	Fast Food, Beverages	Byresandra,Tavarekere,Madiwala	HSR	
2	#refuel	Cafe	3.7	37	400.0	Yes	No	Cafe, Beverages	Bannerghatta Road	Bannerghatta Road	
3	'@ Biryani Central	Casual Dining	2.7	135	550.0	Yes	No	Biryani, Mughlai, Chinese	Marathahalli	Marathahalli	
								BBQ,			

```
from sklearn.preprocessing import MinMaxScaler

# Initialize a scaler
scaler = MinMaxScaler()

# Assuming 'rating' is a numerical column that we want to normalize
df['rating_normalized'] = scaler.fit_transform(df[['rate (out of 5)']])

# Check the transformation
print(df[['rate (out of 5)', 'rating_normalized']].head())
```

```
   rate (out of 5)  rating_normalized
0              3.4           0.516129
1              3.9           0.677419
2              3.7           0.612903
3              2.7           0.290323
4              2.8           0.322581
```

```
# Convert categorical variable 'status' into integer codes
df['cuisines_code'] = df['cuisines type'].astype('category').cat.codes

# Extract unique city and their corresponding codes
unique_cities = df[['cuisines type','cuisines_code']].drop_duplicates().sort_values(by='cuisines type')

# Print the unique cities and their codes
print(unique_cities)
```

```
                                         cuisines type  cuisines_code
2306                                   African, Burger              0
3856                                          American              1
6085  American, Asian, Continental, North Indian, So...             2
4010            American, Asian, European, North Indian              3
2549                                     American, BBQ              4
...                                                ...            ...
7102               Turkish, Fast Food, Biryani, Chinese           2170
6088                                    Turkish, Rolls           2171
882                                         Vietnamese           2172
4705                                 Vietnamese, Salad           2173
6207                Vietnamese, Thai, Burmese, Japanese           2174

[2175 rows x 2 columns]
```

```
import nltk
nltk.download('stopwords')
from nltk.stem import PorterStemmer

ps =PorterStemmer()
import os
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
import nltk
import re
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
import pickle
import seaborn as sns
def preprocess_text(text):
    review=re.sub("[^a-zA-Z]"," ",text)
    review=review.lower()
    review=review.split()
    review=[ps.stem(word) for word in review if word not in set(stopwords.words("english"))]
    review=" ".join(review)
    return review
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
df['restaurant name'] = df['restaurant name'].apply(preprocess_text)
```

```
df
```

	restaurant name	restaurant type	rate (out of 5)	num of ratings	avg cost (two people)	online_order	table booking	cuisines type	area	local address	pc
0	feeltherol	Quick Bites	3.4	7	200.0	No	No	Fast Food	Bellandur	Bellandur	
1	I cafe	Quick Bites	3.9	48	400.0	Yes	No	Fast Food, Beverages	Byresandra,Tavarekere,Madiwala	HSR	
2	refuel	Cafe	3.7	37	400.0	Yes	No	Cafe, Beverages	Bannerghatta Road	Bannerghatta Road	
7102	zu doner kebap	Takeaway, Delivery	3.7	33	350.0	No	No	Turkish, Fast Food, Biryani, Chinese	Malleshwaram	RT Nagar	
7103	zyara	Casual Dining	3.8	191	650.0	Yes	No	North Indian, Mughlai, Chinese	Kammanahalli	HBR Layout	
7104	zyksha	Food Truck	3.4	9	200.0	No	No	Fast Food	Bannerghatta Road	South Bangalore	

7105 rows × 13 columns

```
from sklearn.preprocessing import StandardScaler

# Scale features like 'cost' using Standard Scaler
scaler = StandardScaler()
df['cost_scaled'] = scaler.fit_transform(df[['avg cost (two people)']])

# Check the scaled feature
print(df[['avg cost (two people)', 'cost_scaled']].head())
```

```
   avg cost (two people)  cost_scaled
0                  200.0    -0.735167
1                  400.0    -0.303080
2                  400.0    -0.303080
3                  550.0     0.020985
4                  700.0     0.345051
```

```
Start coding or generate with AI.
```