Restaurant Recommender System using ML

In [37]:

```
# Importing Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import r2_score
from sklearn.metrics import classification_report
import warnings
warnings.filterwarnings('always')
warnings.filterwarnings('ignore')
import re
from nltk.corpus import stopwords
from sklearn.metrics.pairwise import linear_kernel
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [92]:

```
# Reading the dataset
zomato_data = pd.read_csv("/Users/sachin/Downloads/NLP/zomato.csv")
```

In [93]:

```
zomato_data.head(5)
```

Out[93]:

	url	address	name	online_order	book_table
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	Nc
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	Nc
3	https://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	Nc
4	https://www.zomato.com/bangalore/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	Nc

Now the next step is data cleaning and feature engineering for this step we need to do a lot of stuff with the data such as:

Deleting Unnecessary Columns
Removing the Duplicates
Remove the NaN values from the dataset
Changing the column names
Data Transformations
Data Cleaning
Adjust the column names Now, let’s perform all the above steps in our data:

In [94]:

```
# Deleting Unnecessary columns
zomato = zomato_data.drop(["url","dish_liked","phone"],axis=1)
```

In [95]:

```
# Removing the duplicates
zomato.duplicated().sum()  # check number of duplicated rows
zomato.drop_duplicates(inplace=True)
```

In [96]:

```
# Removing NaN
zomato.isnull().sum() # check number of null values in each columns
zomato.dropna(how="any",inplace=True)
```

In [97]:

```
# Renaming Columns
zomato.rename(columns={"approx_cost(for two people)":"cost","listed_ir
```

In [98]:

```
# Transformations
# Removing "," from cost
zomato['cost'] = zomato['cost'].astype(str)
zomato['cost'] = zomato['cost'].apply(lambda x : x.replace(",",""))
zomato['cost'] = zomato['cost'].astype(float)
```

In [99]:

```
# Removing "NEW","-" and "/5" from rate
zomato = zomato.loc[zomato['rate']!="NEW"]
zomato = zomato.loc[zomato['rate']!='-'].reset_index(drop=True)
remove_slash = lambda x: x.replace("/5","") if type(x)==np.str else x
zomato['rate'] = zomato['rate'].apply(remove_slash).str.strip().astype
```

In [104]:

```
# Adjusting the column name
zomato.name = zomato.name.apply(lambda x:x.title())
zomato['book_table'].replace(("Yes","No"),(True,False), inplace=True)
zomato['online_order'].replace(("Yes","No"),(True,False), inplace=True
```

In [114]:

```python
# Computing mean Rating for each restaurant(Feature Engineering)
ratings=pd.DataFrame(zomato.groupby('name')['rate'].mean().reset_index
ratings
```

Out[114]:

	name	rate
0	#Feeltheroll	3.400
1	#L-81 Cafe	3.900
2	#Refuel	3.700
3	1000 B.C	3.200
4	100ÃÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂ°C	3.700
...	...	...
6567	Zoey'S	4.300
6568	Zoroy Luxury Chocolate	4.000
6569	Zu'S Doner Kebaps	3.700
6570	Zyara	3.875
6571	Zyksha	3.400

6572 rows × 2 columns

In [119]:

```python
zomato=pd.merge(zomato, ratings,how='left', on='name' )
zomato.rename(columns={"rate_y":"mean_rate"},inplace=True)
zomato
```

Out[119]:

	address	name	online_order	book_table	rate_x	votes	location	rest_t
0	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	True	True	4.1	775	Banashankari	Ca Di
1	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	True	False	4.1	787	Banashankari	Ca Di
2	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	True	False	3.8	918	Banashankari	C Ca Di
	1st Floor,	Addhuri						

3	Annakuteera, 3rd Stage, Banashankar...	Udupi Bhojana	False	False	3.7	88	Banashankari	Q E
4	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	False	False	3.8	166	Basavanagudi	Ca Di
...	...	...	...	...	...	...	...	
41232	136, SAP Labs India, KIADB Export Promotion In...	The Farm House Bar N Grill	False	False	3.7	34	Whitefield	Ca Din
41233	139/C1, Next To GR Tech Park, Pattandur Agraha...	Bhagini	False	False	2.5	81	Whitefield	Ca Din
41234	Four Points by Sheraton Bengaluru, 43/3, White...	Best Brews - Four Points By Sheraton Bengaluru...	False	False	3.6	27	Whitefield	
41235	Sheraton Grand Bengaluru Whitefield Hotel & Co...	Chime - Sheraton Grand Bengaluru Whitefield Ho...	False	True	4.3	236	ITPL Main Road, Whitefield	
41236	ITPL Main Road, KIADB Export Promotion Industr...	The Nest - The Den Bengaluru	False	False	3.4	13	ITPL Main Road, Whitefield	Ca Di

41237 rows × 16 columns

In [127]:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (1,5))
zomato[['mean_rate']] = scaler.fit_transform(zomato[['mean_rate']]).rc
```

Now the next step is to perform some text preprocessing steps which include:

Lower casing
Removal of Punctuations
Removal of Stopwords
Removal of URLs
Spelling correction
Now let’s perform the above text preprocessing steps on the data:

In [132]:

```
# Lower casing
zomato['reviews_list'] = zomato['reviews_list'].str.lower()

# Removal of Punctuations
import string
punc_to_remove = string.punctuation
def remove_punctuations(text):
    return text.translate(str.maketrans('', '', punc_to_remove))

zomato['reviews_list'] =zomato['reviews_list'].apply(lambda text: remo
```

In [137]:

```
# Removal of Stopwords
stop_words = set(stopwords.words('english'))
def remove_stopwords(text):
    return " ".join([word for word in str(text).split() if word not in
zomato['reviews_list'] = zomato['reviews_list'].apply(lambda text: rem
```

In [140]:

```
# Removal of URL
def remove_url(text):
    url_pattern = re.compile(r'https?://\S+|www\.\S+')
    return url_pattern.sub(r'',text)
zomato['reviews_list'] = zomato['reviews_list'].apply(lambda text: rem
```

In [141]:

```
zomato['reviews_list']
```

Out[141]:

```
0        rated 40 ratedn beautiful place dine inthe int...
1        rated 40 ratedn dinner family turned good choo...
2        rated 30 ratedn ambience good enough pocket fr...
3        rated 40 ratedn great food proper karnataka st...
4        rated 40 ratedn good restaurant neighbourhood ...
                                ...
41232    rated 40 ratedn ambience big spacious lawn use...
41233    rated 40 ratedn fine place chill office hours ...
41234    rated 50 ratedn food service incomparably exce...
41235    rated 40 ratedn nice friendly place staff awes...
41236    rated 50 ratedn great ambience looking nice go...
Name: reviews_list, Length: 41237, dtype: object
```

TF-IDF Vectorizer

```
In [143]: zomato.drop(columns=['address','rest_type', 'type', 'menu_item', 'vote
          df_percent.rename(columns={'mean_rate':'Mean Rating'},inplace=True)
```

```
In [145]: # Randomly sampling 60% OF dataset
          df_percent = zomato.sample(frac=0.6)
```

```
In [149]: df_percent.set_index('name',inplace=True)
          indices = pd.Series(df_percent.index)
```

```
In [153]: # Creating TF-IDF Matrix
          tfidf = TfidfVectorizer(analyzer='word',ngram_range=(1,2),min_df=0, st
          tfidf_matrix = tfidf.fit_transform(df_percent['reviews_list'])
          cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
```

```
In [156]: cosine_similarities
```

```
Out[156]: array([[1.00000000e+00, 2.77366910e-03, 1.07108322e-02, ...,
                  5.30357196e-03, 2.36064866e-02, 6.79103198e-03],
                 [2.77366910e-03, 1.00000000e+00, 4.93025119e-03, ...,
                  6.34095547e-04, 1.13662745e-02, 2.69431239e-03],
                 [1.07108322e-02, 4.93025119e-03, 1.00000000e+00, ...,
                  4.72002043e-03, 2.75998475e-02, 1.98241294e-02],
                 ...,
                 [5.30357196e-03, 6.34095547e-04, 4.72002043e-03, ...,
                  1.00000000e+00, 7.74911196e-03, 3.07933572e-03],
                 [2.36064866e-02, 1.13662745e-02, 2.75998475e-02, ...,
                  7.74911196e-03, 1.00000000e+00, 2.52733138e-02],
                 [6.79103198e-03, 2.69431239e-03, 1.98241294e-02, ...,
                  3.07933572e-03, 2.52733138e-02, 1.00000000e+00]])
```

Building A Recommender System

In [169]:
```
def recommend(name, cosine_similarities=cosine_similarities):
    recommend_restaurant = [] # list to append recommended restaurant
    idx = indices[indices==name].index[0]  # Getting the index of rest
    score_series = pd.Series(cosine_similarities[idx]).sort_values(asc
    top_30indices = list(score_series.iloc[0:31].index) # extracting t

    for val in top_30indices:
        recommend_restaurant.append(list(df_percent.index)[val])

    # Creating a new DF to shown similary restaurants
    df_new = pd.DataFrame(columns=['cuisines', 'Mean Rating', 'cost'])

    for each in recommend_restaurant:
        df_new = df_new.append(pd.DataFrame(df_percent[['cuisines','Me

    # Dropping Duplicates
    df_new = df_new.drop_duplicates(subset=['cuisines','Mean Rating',
    # sorting and keeping only top 10 values
    df_new = df_new.sort_values(by='Mean Rating',ascending=False).heac

    # Printing
    print('TOP %s RESTAURANTS LIKE %s WITH SIMILAR REVIEWS: ' % (str(l

    return df_new
```

```
In [170]: recommend('Pai Vihar')
```

```
TOP 10 RESTAURANTS LIKE Pai Vihar WITH SIMILAR REVIEWS:
```

Out[170]:

	cuisines	Mean Rating	cost
Burma Burma	Asian, Burmese	4.74	1500.0
Lavonne	Cafe, Desserts	4.35	800.0
CafãÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂÃÂ© Felix	American, Cafe, Continental	4.35	1700.0
Foxtrot - House Of Subculture	Cafe, American, Asian, North Indian	4.35	1000.0
The Lantern Restaurant & Bar - The Ritz-Carlton...	Chinese	4.28	3500.0
Marzipan Cafe & Bakery	Cafe, Mediterranean, Bakery, Greek, Beverages	4.10	700.0
Ilyazsab The House Of Chicken	Rolls, Kebab	3.84	250.0
Foodhall	Italian, Bakery, Fast Food	3.80	1000.0
1992 Chats - Space	Street Food	3.45	200.0
Karavali Grand	Mangalorean, Seafood, North Indian, Chinese	3.45	600.0