Capstone Project - 2 : Zomato Recommendation System (NLP & Recommender system)

Introduction:

The purpose of this project is to determine what makes a good restaurant and build a restaurant recommender system to make the task of choosing a proper place a bit easier.

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import r2_score
import warnings
warnings.filterwarnings("ignore")
import re
import nltk
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 [2]:
```
df = pd.read_csv("zomato.csv")
df
```

Out[2]:

	url	address	name	online_order	book_table	rate	votes	phone	location	res
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	080 42297555\r\n+91 9743772233	Banashankari	
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	080 41714161	Banashankari	
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	+91 9663487993	Banashankari	
3	https://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 9620009302	Banashankari	
4	https://www.zomato.com/bangalore/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	+91 8026612447\r\n+91 9901210005	Basavanagudi	
...	...	...	...	...	...	...	...	...	...	
51712	https://www.zomato.com/bangalore/best-brews-fo...	Four Points by Sheraton Bengaluru, 43/3, White...	Best Brews - Four Points by Sheraton Bengaluru...	No	No	3.6 /5	27	080 40301477	Whitefield	
51713	https://www.zomato.com/bangalore/vinod-bar-and...	Number 10, Garudachar Palya, Mahadevapura, Whi...	Vinod Bar And Restaurant	No	No	NaN	0	+91 8197675843	Whitefield	
51714	https://www.zomato.com/bangalore/plunge-sherat...	Sheraton Grand Bengaluru Whitefield Hotel & Co...	Plunge - Sheraton Grand Bengaluru Whitefield H...	No	No	NaN	0	NaN	Whitefield	
51715	https://www.zomato.com/bangalore/chime-sherato...	Sheraton Grand Bengaluru Whitefield Hotel & Co...	Chime - Sheraton Grand Bengaluru Whitefield Ho...	No	Yes	4.3 /5	236	080 49652769	ITPL Main Road, Whitefield	
51716	https://www.zomato.com/bangalore/the-nest-the-...	ITPL Main Road, KIADB Export Promotion Industr...	The Nest - The Den Bengaluru	No	No	3.4 /5	13	+91 8071117272	ITPL Main Road, Whitefield	

51717 rows × 17 columns

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51717 entries, 0 to 51716
Data columns (total 17 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   url                          51717 non-null  object
 1   address                      51717 non-null  object
 2   name                         51717 non-null  object
 3   online_order                 51717 non-null  object
 4   book_table                   51717 non-null  object
 5   rate                         43942 non-null  object
 6   votes                        51717 non-null  int64 
 7   phone                        50509 non-null  object
 8   location                     51696 non-null  object
 9   rest_type                    51490 non-null  object
 10  dish_liked                   23639 non-null  object
 11  cuisines                     51672 non-null  object
 12  approx_cost(for two people)  51371 non-null  object
 13  reviews_list                 51717 non-null  object
```

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
url                                0
address                            0
name                               0
online_order                       0
book_table                         0
rate                            7775
votes                              0
phone                           1208
location                          21
rest_type                        227
dish_liked                     28078
cuisines                          45
approx_cost(for two people)      346
reviews_list                       0
menu_item                          0
listed_in(type)                    0
listed_in(city)                    0
dtype: int64
```

EDA, Data Cleaning and Feature Engineering

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 [5]:
```
#Dropping the column "dish_liked", "phone", "url" and saving the new dataset as "zomato"
zomato=df.drop(['url','phone','dish_liked'],axis=1)
print(zomato.head())
```

```
                                             address                  name  \
0  942, 21st Main Road, 2nd Stage, Banashankari, ...                 Jalsa   
1  2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...        Spice Elephant   
2  1112, Next to KIMS Medical College, 17th Cross...       San Churro Cafe   
3  1st Floor, Annakuteera, 3rd Stage, Banashankar...  Addhuri Udupi Bhojana   
4  10, 3rd Floor, Lakshmi Associates, Gandhi Baza...         Grand Village   

  online_order book_table   rate  votes      location            rest_type  \
0          Yes        Yes  4.1/5    775  Banashankari        Casual Dining   
1          Yes         No  4.1/5    787  Banashankari        Casual Dining   
2          Yes         No  3.8/5    918  Banashankari  Cafe, Casual Dining   
3           No         No  3.7/5     88  Banashankari          Quick Bites   
4           No         No  3.8/5    166  Basavanagudi        Casual Dining   

                     cuisines approx_cost(for two people)  \
0  North Indian, Mughlai, Chinese                         800   
1     Chinese, North Indian, Thai                         800   
2          Cafe, Mexican, Italian                         800   
3      South Indian, North Indian                         300   
4        North Indian, Rajasthani                         600   

                                        reviews_list menu_item  \
0  [('Rated 4.0', 'RATED\n  A beautiful place to ...        []   
1  [('Rated 4.0', 'RATED\n  Had been here for din...        []   
2  [('Rated 3.0', "RATED\n  Ambience is not that ...        []   
3  [('Rated 4.0', "RATED\n  Great food and proper...        []   
4  [('Rated 4.0', 'RATED\n  Very good restaurant ...        []   

  listed_in(type) listed_in(city)  
0          Buffet    Banashankari  
1          Buffet    Banashankari  
2          Buffet    Banashankari  
3          Buffet    Banashankari  
4          Buffet    Banashankari  
```

In [6]:
```
#Removing the Duplicates
zomato.duplicated().sum()
zomato.drop_duplicates(inplace=True)
```

In [7]:
```
print("Shape before removing duplicates:", zomato.shape)
zomato.drop_duplicates(inplace=True)
print("Shape after removing duplicates:", zomato.shape)
```

```
Shape before removing duplicates: (51674, 14)
Shape after removing duplicates: (51674, 14)
```

In [8]:
```
#Removing the NaN values from the dataset
zomato.isnull().sum()
zomato.dropna(how='any',inplace=True)
print("Shape before removing NaN values:", zomato.shape)
zomato.dropna(how='any', inplace=True)
print("Shape after removing NaN values:", zomato.shape)
```

```
Shape before removing NaN values: (43499, 14)
Shape after removing NaN values: (43499, 14)
```

In [9]:
```
#Changing the column names
zomato = zomato.rename(columns={'approx_cost(for two people)':'cost','listed_in(type)':'type', 'listed_in(city)':'city'})
print(zomato.columns)
```

```
Index(['address', 'name', 'online_order', 'book_table', 'rate', 'votes',
       'location', 'rest_type', 'cuisines', 'cost', 'reviews_list',
       'menu_item', 'type', 'city'],
      dtype='object')
```

In [10]:
```
#Transformations
zomato['cost'] = zomato['cost'].astype(str) #Changing the cost to string
zomato['cost'] = zomato['cost'].apply(lambda x: x.replace(',','.')) #Using lambda function to replace ',' from cost
zomato['cost'] = zomato['cost'].astype(float)
print(zomato['cost'].dtype)
```

```
float64
```

In [11]:
```
# Removing '/5' from Rates
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) == str else x
zomato['rate'] = zomato['rate'].apply(remove_slash).str.strip().astype(float)
print(zomato['rate'].dtype)
print(zomato['rate'].head())
```

```
float64
0    4.1
1    4.1
2    3.8
3    3.7
4    3.8
Name: rate, dtype: float64
```

In [12]:
```
# Adjust the column names
zomato.name = zomato.name.apply(lambda x:x.title())
zomato.online_order.replace(('Yes','No'),(True, False),inplace=True)
zomato.book_table.replace(('Yes','No'),(True, False),inplace=True)
print(zomato['name'].head())
print(zomato['online_order'].head())
print(zomato['book_table'].head())
```

```
0                   Jalsa
1          Spice Elephant
2         San Churro Cafe
3    Addhuri Udupi Bhojana
4           Grand Village
Name: name, dtype: object
0     True
1     True
2     True
3    False
4    False
Name: online_order, dtype: bool
0     True
1    False
2    False
3    False
4    False
Name: book_table, dtype: bool
```

In [13]:
```
## Computing Mean Rating
restaurants = list(zomato['name'].unique())
zomato['Mean Rating'] = 0

for i in range(len(restaurants)):
    zomato['Mean Rating'][zomato['name'] == restaurants[i]] = zomato['rate'][zomato['name'] == restaurants[i]].mean()

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (1,5))
zomato[['Mean Rating']] = scaler.fit_transform(zomato[['Mean Rating']]).round(2)
print(zomato['Mean Rating'].head())
```

```
0    3.99
1    3.97
2    3.58
3    3.45
4    3.58
Name: Mean Rating, dtype: float64
```

Now in the next step, we perform text preprocessing steps:

Lower casing

Removal of Punctuations

Removal of Stopwords

Removal of URLs

Spelling correction

In [14]:
```
## Lower Casing
zomato["reviews_list"] = zomato["reviews_list"].str.lower()
print(zomato["reviews_list"].head())
```

```
0    [('rated 4.0', 'rated\n  a beautiful place to ...
1    [('rated 4.0', 'rated\n  had been here for din...
2    [('rated 3.0', "rated\n  ambience is not that ...
3    [('rated 4.0', "rated\n  great food and proper...
4    [('rated 4.0', 'rated\n  very good restaurant ...
Name: reviews_list, dtype: object
```

In [15]:

```
# Defining a function to remove punctuation using regular expressions
def remove_punctuation(text):
    """Remove punctuation from the text"""
    return re.sub(r'[^\w\s]', '', text)

# Applying the function to the "reviews_list" column
zomato["reviews_list"] = zomato["reviews_list"].apply(remove_punctuation)
print(zomato["reviews_list"].head())
```

```
0    rated 40 ratedn  a beautiful place to dine int...
1    rated 40 ratedn  had been here for dinner with...
2    rated 30 ratedn  ambience is not that good eno...
3    rated 40 ratedn  great food and proper karnata...
4    rated 40 ratedn  very good restaurant in neigh...
Name: reviews_list, dtype: object
```

In [16]:

```
STOPWORDS = set(stopwords.words('english'))
def remove_stopwords(text):
    """custom function to remove the stopwords"""
    return " ".join([word for word in str(text).split() if word not in STOPWORDS])

zomato["reviews_list"] = zomato["reviews_list"].apply(lambda text: remove_stopwords(text))
```

In [17]:

```
## Removal of URLS
def remove_urls(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: remove_urls(text))

zomato[['reviews_list', 'cuisines']].sample(5)
```

Out[17]:

	reviews_list	cuisines
32548	rated 40 ratedn restaurant really nice really ...	Fast Food, Chinese, Burger, Hot dogs
33204	rated 10 ratedn pathetic drinks ordered two mi...	North Indian, Chinese
21737	rated 10 ratedn place doesnt understand meanin...	Cafe, Continental, Burger
20820	rated 50 ratedn yummy pocket friendly n satisf...	Tea, Coffee, Fast Food
19996	rated 40 ratedn ordered food late night zomato...	Fast Food

In [18]:

```
# RESTAURANT NAMES:
restaurant_names = list(zomato['name'].unique())
def get_top_words(column, top_nu_of_words, nu_of_word):
    vec = CountVectorizer(ngram_range= nu_of_word, stop_words='english')
    bag_of_words = vec.fit_transform(column)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]
    words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
    return words_freq[:top_nu_of_words]

zomato=zomato.drop(['address','rest_type', 'type', 'menu_item', 'votes'],axis=1)
```

In [19]:

```
# Randomly sample 60% of your dataframe
df_percent = zomato.sample(frac=0.5)
```

In [20]:

```
zomato.head()
```

Out[20]:

	name	online_order	book_table	rate	location	cuisines	cost	reviews_list	city	Mean Rating
0	Jalsa	True	True	4.1	Banashankari	North Indian, Mughlai, Chinese	800.0	rated 40 ratedn beautiful place dine inthe int...	Banashankari	3.99
1	Spice Elephant	True	False	4.1	Banashankari	Chinese, North Indian, Thai	800.0	rated 40 ratedn dinner family turned good choo...	Banashankari	3.97
2	San Churro Cafe	True	False	3.8	Banashankari	Cafe, Mexican, Italian	800.0	rated 30 ratedn ambience good enough pocket fr...	Banashankari	3.58
3	Addhuri Udupi Bhojana	False	False	3.7	Banashankari	South Indian, North Indian	300.0	rated 40 ratedn great food proper karnataka st...	Banashankari	3.45
4	Grand Village	False	False	3.8	Basavanagudi	North Indian, Rajasthani	600.0	rated 40 ratedn good restaurant neighbourhood ...	Banashankari	3.58

In [21]:

```
zomato.sample(5)
```

Out[21]:

	name	online_order	book_table	rate	location	cuisines	cost	reviews_list	city	Mean Rating
5098	Mughals Restaurant	False	False	3.2	Shivajinagar	Mughlai, North Indian, Chinese, Biryani, Seafood	550.0	rated 50 ratedn good food quit reasonable must...	Brigade Road	2.74
22252	Trigereato	True	True	4.4	Koramangala 8th Block	Cafe, Continental, Beverages	700.0	rated 45 ratedn crowdy sunday night invites tr...	Koramangala 4th Block	4.34
8969	Biryani Foodies	True	False	3.1	Koramangala 5th Block	Biryani, Mughlai	500.0	rated 10 ratedn ordered panneer biryani worst ...	BTM	2.68
35882	Roll Er Dokaan	True	False	3.9	Jeevan Bhima Nagar	Street Food, Rolls	300.0	rated 10 ratedn absolutely stale poor quality ...	Old Airport Road	3.02
27689	Zero Mile Punjab	True	True	4.1	HSR	North Indian, Mughlai	800.0	rated 20 ratedn went dinner friends ordered pa...	Koramangala 7th Block	3.97

In [22]:
```
zomato.shape
```

Out[22]:
```
(41237, 10)
```

In [23]:
```
zomato.columns
```

Out[23]:
```
Index(['name', 'online_order', 'book_table', 'rate', 'location', 'cuisines',
       'cost', 'reviews_list', 'city', 'Mean Rating'],
      dtype='object')
```

In [24]:
```
df_percent.shape
```

Out[24]:
```
(20618, 10)
```

TF-IDF Vectorization

In [25]:
```
df_percent.set_index('name', inplace=True)
indices = pd.Series(df_percent.index)
```

In [26]:
```
# Creating tf-idf matrix
tfidf = TfidfVectorizer(analyzer='word', ngram_range=(1, 2), min_df=1, stop_words='english')
tfidf_matrix = tfidf.fit_transform(df_percent['reviews_list'])

cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
```

Now the last step for creating a Restaurant Recommendation System is to write a function that will recommend restaurants:

In [27]:
```
def recommend(name, cosine_similarities = cosine_similarities):

    # Create a list to put top restaurants
    recommend_restaurant = []

    # Find the index of the hotel entered
    idx = indices[indices == name].index[0]

    # Find the restaurants with a similar cosine-sim value and order them from bigges number
    score_series = pd.Series(cosine_similarities[idx]).sort_values(ascending=False)

    # Extract top 30 restaurant indexes with a similar cosine-sim value
    top30_indexes = list(score_series.iloc[0:31].index)

    # Names of the top 30 restaurants
    for each in top30_indexes:
        recommend_restaurant.append(list(df_percent.index)[each])

    # Creating the new data set to show similar restaurants
    df_new = pd.DataFrame(columns=['cuisines', 'Mean Rating', 'cost'])

    # Create the top 30 similar restaurants with some of their columns
    for each in recommend_restaurant:
        df_new = df_new.append(pd.DataFrame(df_percent[['cuisines','Mean Rating', 'cost']][df_percent.index == each].sample

    # Drop the same named restaurants and sort only the top 10 by the highest rating
    df_new = df_new.drop_duplicates(subset=['cuisines','Mean Rating', 'cost'], keep=False)
    df_new = df_new.sort_values(by='Mean Rating', ascending=False).head(10)

    print('TOP %s RESTAURANTS LIKE %s WITH SIMILAR REVIEWS: ' % (str(len(df_new)), name))

    return df_new
recommend('Pai Vihar')
```

```
TOP 4 RESTAURANTS LIKE Pai Vihar WITH SIMILAR REVIEWS:
```

Out[27]:

	cuisines	Mean Rating	cost
Swad Punjab Da	North Indian	3.87	150.0
Kakaji	North Indian	3.45	350.0
Prasiddhi Food Corner	Fast Food, North Indian, South Indian	3.45	200.0
Mayura Sagar	Chinese, North Indian, South Indian	3.32	250.0

In [ ]:

In this project, we developed a content-based restaurant recommender system using Zomato data. We began by preprocessing the data, including handling missing values, text normalization, and feature engineering. Employing TF-IDF and cosine similarity, we successfully recommended similar restaurants based on user-entered preferences. While the system provides promising recommendations, ongoing refinement and evaluation are crucial for ensuring fairness, relevance, and user satisfaction. Further enhancements could involve incorporating user feedback mechanisms and exploring collaborative filtering techniques for more personalized recommendations.

Submitted by- Shweta Kanungo

In [ ]: