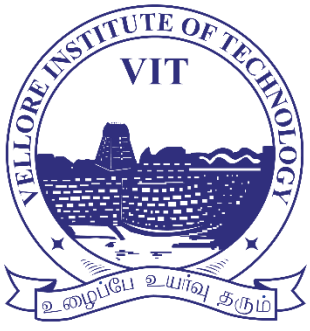
Restaurant Recommendation System

Name:-Harshal Jain(21BCI0119)

Devashish Sachdeva(21BCE0243)

Siddhant Chamoli (21BCE2140)

Arnav (21BCE3582)



VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Video Link:-
https://drive.google.com/drive/folders/19hyRjBs9WwM26d75s1-nz-SQMaAcr1qH?usp=sharing

Performance Measure: The evaluation criteria for an AI-driven restaurant and cuisine recommendation system would primarily focus on user satisfaction. This could be gauged by the accuracy of suggestions aligning with users' tastes, the variety of recommendations provided, the system's efficiency in delivering suggestions promptly, and the overall user engagement and loyalty. Additionally, monitoring the frequency of successful restaurant visits or orders facilitated by the system can also serve as an indicator of its performance.

Environment: The setting for this AI agent encompasses various components, including user input via a digital platform like a website or mobile application, a repository of restaurants and culinary options, and potentially dynamic data such as current trends, user feedback, and

location-specific information. The system interacts with users to grasp their preferences, dietary needs, financial considerations, and any other pertinent factors influencing their dining decisions. It then processes this input to generate tailored recommendations suited to each user's requirements.

Actuators: The mechanisms involved in this environment primarily revolve around the digital interface through which the AI agent communicates with users. This might encompass text-based exchanges, voice commands, or a blend of both. The system also interfaces with external platforms such as online reservation systems or food delivery services to enable users to act upon the recommendations provided. Additionally, it may employ notification systems to keep users informed about new suggestions or restaurant promotions.

Sensors: The sensors within this context capture user input and feedback, including text inputs,

voice commands, ratings, reviews, and behavioral data like click-through rates and interaction patterns. Continuously gathering and analyzing this data helps enhance the system's recommendation algorithms, adapt to evolving user preferences, and refine the overall user experience. It may also integrate with external application programming interfaces (APIs) to fetch real-time information regarding restaurant availability, menu updates, and other pertinent factors shaping recommendations.

1. INTRODUCTION

1.1 Project Overview:-

A restaurant recommendation system (RRS) is the focus of this research. An information filtering system called a recommendation system makes an attempt to forecast the rating a user would assign to the item—in this case, a restaurant. RRS is an online restaurant search system. All Bangalore's restaurants are available for browsing. Obtain details about the name, kind, rating, and cost of the restaurant. Among the features are restaurant searches and recommendation viewing. Systems that provide recommendations are essential for boosting sales for businesses and enabling customers to locate eateries that suit their preferences. The fact that so many users don't rate the restaurants and users that are introduced to the system every day makes it difficult to use. We

must forecast the rating for the restaurants that are not rated in order to enhance the restaurant rating system. Therefore, developing a recommendation system for restaurants with low ratings is crucial. Users only need to enter the name of a restaurant they have enjoyed visiting in the past into this recommendation algorithm, and it will produce a list of the top 10 restaurants based on the highest cosine similarity scores to that specific restaurant. In order to offer restaurants that match a user's interests, the content-based recommendation methodology suggests restaurants to consumers based on related restaurant categories and popular topic keywords.

1.2 Purpose

This system's goal is to give consumers suggestions for restaurants that would be best for them. People can acquire suggestions from this method, and you can also get other people's viewpoints via this website. Additionally, you can browse the ratings page, which compiles feedback and experiences from numerous individuals, to identify the top eateries. The idea behind this system is that users may browse through the data you send and find all the restaurants that have responded to consumers' requests. This system functions similarly to a bulletin board for foodies. Customers must use this website to look up restaurants by name. They will get a page describing the related names of the restaurants and their type and ratings.

1.3 Problem Statement Definition

As we are users of recommendation applications, people care more about how we will like a restaurant. It is very common that we hang out with families, friends, and co-workers. when comes to lunch or dinner time. In the past, people obtained suggestions for restaurants from friends. Although this method is straightforward and user-

friendly, it has some severe limitations. First, the recommendations from friends or other common people are limited to those places they have visited before. Thus, the user is not able to gain information about places less visited by their friends. Besides that, there is a chance of users not liking the place recommended by their friends. So our project gives a way to user to find similar restaurants to the restaurants they already like without asking for suggestions from their friends or family.

REQUIREMENT ANALYSIS

Functional Requirement

Hardware and Software Software Requirements:

To complete this project, you will require the following software's, concepts, and packages Anaconda navigator Python packages:

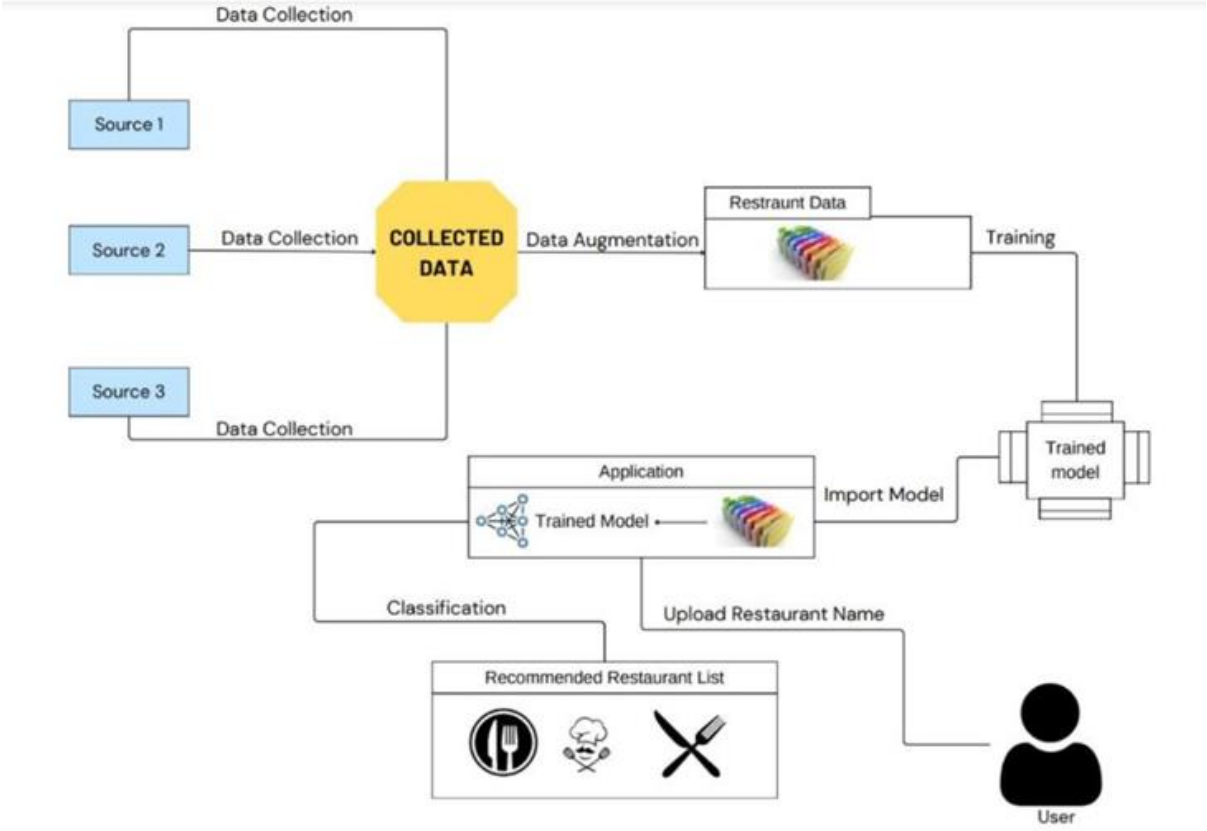
- pandas
- matplotlib
- seaborn
- plotly
- numpy
- scikit-image
- scikit-learn
- Flask 4.2

Non-Functional Requirements Hardware Requirements

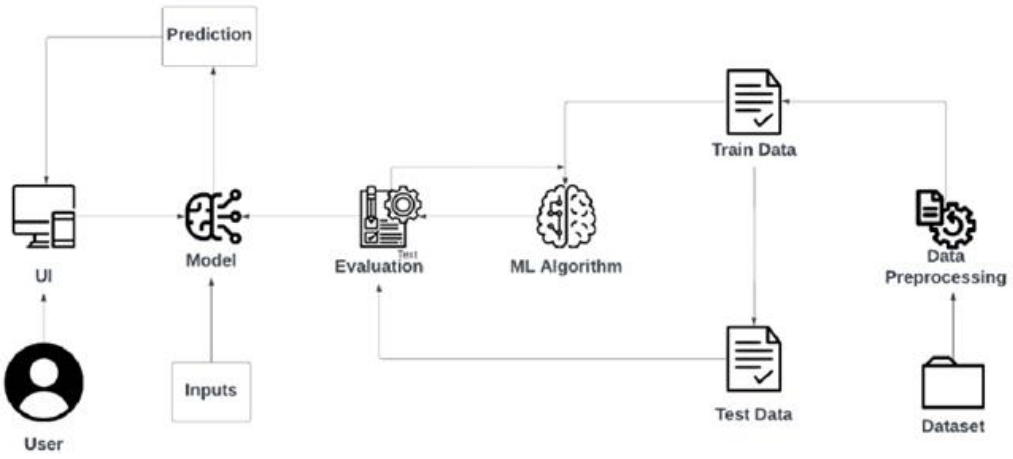
- Processor : Intel Core i3
- Hard Disk Space : Min 100 GB
- Ram : 4 GB

Other than this we will also need the user to enter a restaurant according to which our recommendation system will make the recommendations.

Data Flow Diagram:



Solution Architecture Diagram:-



Workflow:

• User inputs a restaurant name via the web interface.

• The Flask app processes the input and interacts with the recommendation model.

• Text data from the Zomato dataset undergoes TF-IDF vectorization.

• Cosine similarity calculation to find similar restaurants based on reviews.

• Top similar restaurants are selected and displayed to the user via the web interface.

Technologies Used:

• Python (Flask, Pandas, NumPy, Seaborn, Matplotlib, Plotly, Scikit-learn)

• Pickle for model serialization

• Web development: HTML/CSS, Jinja templating

• NLP libraries: NLTK for text processing

This architecture supports a user-friendly interface where users can discover similar restaurants based on their preferences and explores a solution using text-based analysis for recommendations.

Algorithms

1. CountVectorizer: This algorithm is used for converting a collection of text documents into a matrix of token counts. In the code, it's used to analyze the restaurant reviews. Here's a breakdown of what it does:

- **Tokenization:** Breaks down the reviews into individual words or phrases (tokens).
- **Stop Word Removal:** Removes commonly used words (like "the", "a", "an") that don't hold much meaning for recommendation purposes.
- **Counting:** Creates a matrix where rows represent documents (reviews) and columns represent unique tokens. Each cell contains the count of a specific word appearing in a particular review.

2. TfidfVectorizer: This builds on CountVectorizer but instead of simple word counts, it uses a weight to reflect the importance of a word. Words that appear frequently across all documents get lower weights, while words that are unique or specific to a review get higher weights. This helps identify keywords that better distinguish restaurants.

3. MinMaxScaler: This algorithm scales the "Mean Rating" column between a range of 1 to 5. This is useful because some distance or similarity metrics used later work better with normalized data.

4. Cosine Similarity: This is a metric used to measure how similar two documents (reviews) are. It calculates the cosine of the angle between two vectors representing the documents in the TF-IDF matrix. Higher cosine similarity indicates more similar reviews.

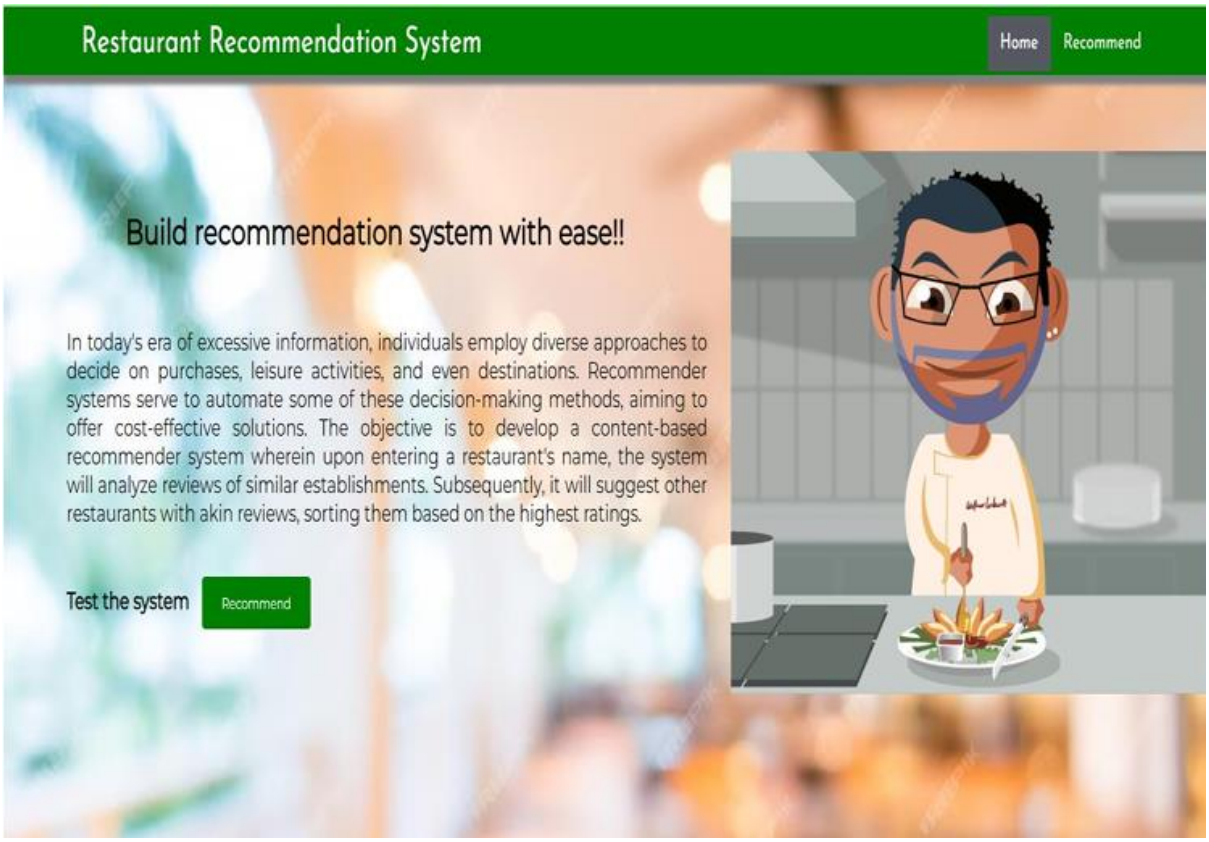
5. Linear Kernel: This is a mathematical function used in conjunction with cosine similarity. It efficiently calculates the cosine similarity between two documents (reviews) based on their TF-IDF vectors.

In summary:

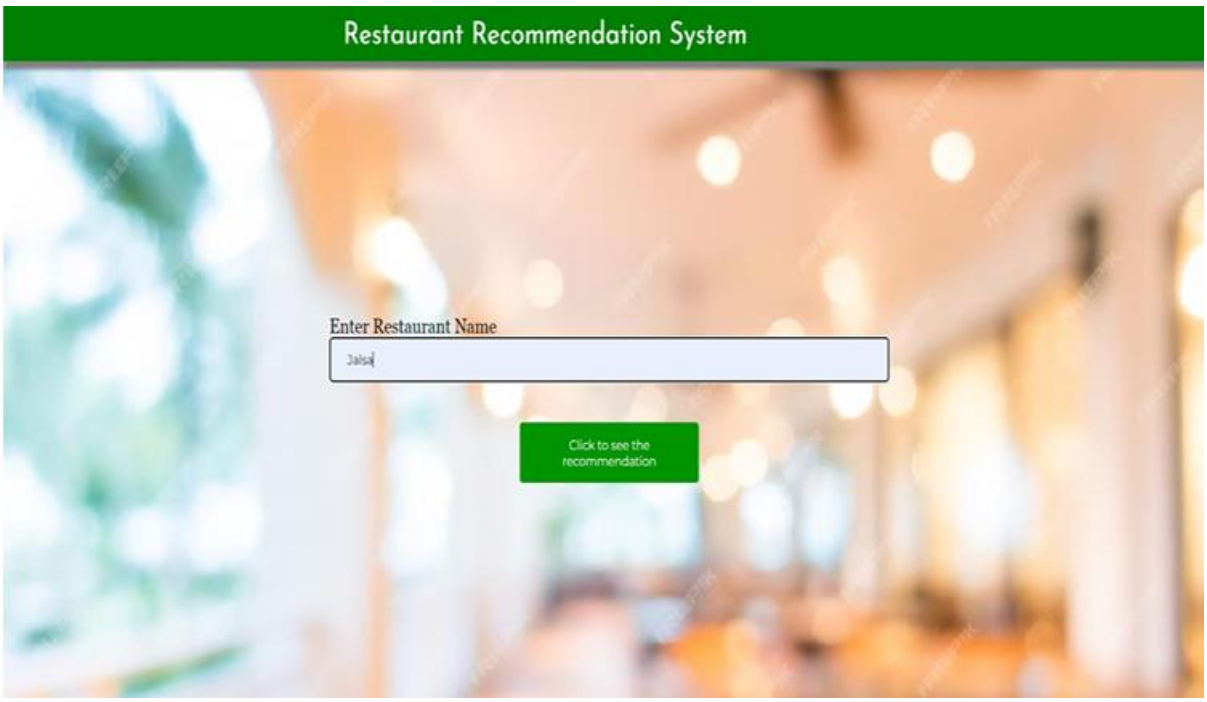
- The code prepares the data for recommendation by cleaning reviews, converting them to numerical features, and calculating the importance of words.
- It then uses cosine similarity to find restaurants with reviews similar to a chosen restaurant, effectively recommending places users might enjoy based on past review patterns.

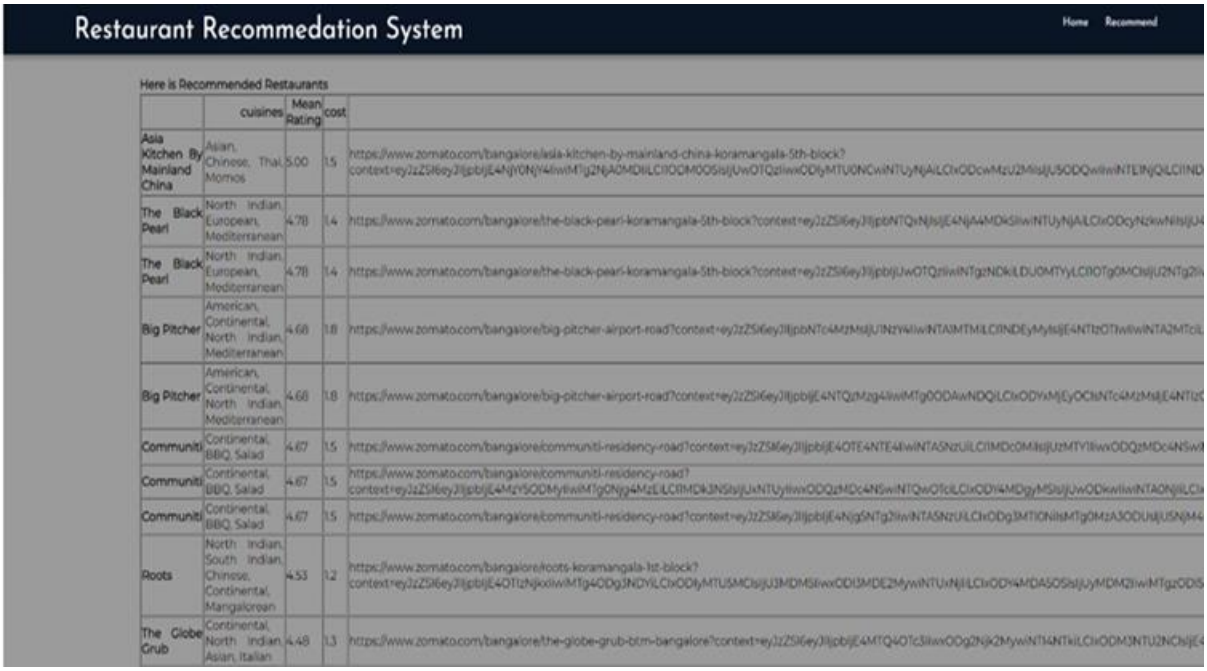
Result

This is the home main page that describes the project and summarizes it.



Checking recommendation for the restaurant: 'Jalsa'





This is the page for predictions, where we input a restaurant's name to receive the top recommended restaurants. Recommendations are based on factors like cuisines, average rating (on a scale of 5), and cost in thousands.

Conclusion

The primary aim of this research is to create an innovative restaurant recommendation system by leveraging machine learning integrated with a user-friendly web interface, essentially functioning as an application designed for customers. This application is intended to assist users in predicting suitable restaurants and identifying the most popular dishes based on geographical regions and individual preferences. It ensures that customers have access to restaurant ratings. By employing both popularity-based and collaborative-based filtering techniques, the recommendation system is optimized to enhance efficiency, enabling every user to effortlessly find suitable restaurants. One common scenario often involves users seeking restaurants in close proximity to their current location. We're addressing this requirement by incorporating restaurant locations

into our dataset. By doing so, our machine learning algorithm becomes adept at predicting suitable restaurants for customers based on their present location. The envisioned restaurant recommendation system, implemented as a web application, is designed to significantly enhance the user experience when searching for restaurants. Its main focus is to provide efficient and rapid restaurant suggestions based on proximity, thereby reducing user effort and optimizing time management. This innovative system not only forecasts appropriate restaurants but also offers insights into popular regional dishes, catering to individual tastes. By leveraging machine learning algorithms, the application ensures that users receive personalized and tailored recommendations. The use of popularity-based and collaborative based filtering techniques serves to refine and optimize the accuracy of suggestions provided to users. Moreover, the integration of location data within the dataset significantly improves the system's predictive capabilities. This allows for precise recommendations, considering a user's present location, thus saving time and effort that would otherwise be spent in manually searching for nearby restaurants. The fundamental objective is to streamline the process of finding a restaurant by empowering users with a user-friendly and efficient tool. By simplifying the search for a dining establishment, users can spend less time browsing through various options and, instead, rely on the system's accurate and personalized suggestions. This expedites the decision-making process for the user, consequently making their time more valuable and saving them from unnecessary hassle. In essence, this comprehensive system aims to revolutionize the way users search for and select restaurants by introducing an intelligent, user-centric approach. It seeks to diminish the complexities of decision-making in restaurant selection and contribute to an enhanced dining experience for users.

restaurant2

April 28, 2024

[1]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[2]:
```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
import plotly.offline as py
import plotly.graph_objs as go
import seaborn as sns
import warnings
warnings.filterwarnings('always')
warnings.filterwarnings('ignore')
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
```

[3]:
```
zomato_data=pd.read_csv("/content/drive/MyDrive/Colab Notebooks/zomato.csv")
zomato_df=zomato_data.copy()
zomato_df.head(2)
```

[3]:
```
                                                 url   \
0  https://www.zomato.com/bangalore/jalsa-banasha…
1  https://www.zomato.com/bangalore/spice-elephan…

                                             address             name  \
0  942, 21st Main Road, 2nd Stage, Banashankari, …            Jalsa
1  2nd Floor, 80 Feet Road, Near Big Bazaar, 6th …  Spice Elephant

  online_order book_table   rate  votes                          phone  \
0          Yes        Yes  4.1/5    775  080 42297555\r\n+91 9743772233
1          Yes         No  4.1/5    787                   080 41714161

       location      rest_type  \
```

```
0  Banashankari  Casual Dining
1  Banashankari  Casual Dining

                                          dish_liked  \
0  Pasta, Lunch Buffet, Masala Papad, Paneer Laja…
1  Momos, Lunch Buffet, Chocolate Nirvana, Thai G…

                        cuisines approx_cost(for two people)  \
0  North Indian, Mughlai, Chinese                         800
1     Chinese, North Indian, Thai                         800

                                       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…        []

  listed_in(type) listed_in(city)
0          Buffet    Banashankari
1          Buffet    Banashankari
```

[4]:
```
zomato_df.shape
```

[4]:
```
(51717, 17)
```

[5]:
```
zomato_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
 14  menu_item                    51717 non-null  object
 15  listed_in(type)              51717 non-null  object
 16  listed_in(city)              51717 non-null  object
```

```
dtypes: int64(1), object(16)
memory usage: 6.7+ MB
```

[6]:
```
zomato_df.isnull().sum()
```

[6]:
```
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
```

[7]:
```
#Dropping the column "dish_liked", "phone", "url"
zomato_df=zomato_df.drop(['phone','dish_liked'],axis=1)

#Remove the NaN values from the dataset
zomato_df.dropna(how='any',inplace=True)

#Removing the Duplicates
zomato_df.duplicated().sum()
zomato_df.drop_duplicates(inplace=True)

#Changing the column names
zomato_df = zomato_df.rename(columns={'approx_cost(for two people)':
 ↪'cost','listed_in(type)':'type', 'listed_in(city)':'city'})

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

#Changing the cost to string
zomato_df['cost'] = zomato_df['cost'].astype(str)
zomato_df['cost'] = zomato_df['cost'].apply(lambda x: x.replace(',','.'))
```

```
zomato_df['cost'] = zomato_df['cost'].astype(float)
```

[8]:
```
zomato_df.shape
```

[8]:
```
(41263, 15)
```

[9]:
```
zomato_df.isnull().sum()
```

[9]:
```
url             0
address         0
name            0
online_order    0
book_table      0
rate            0
votes           0
location        0
rest_type       0
cuisines        0
cost            0
reviews_list    0
menu_item       0
type            0
city            0
dtype: int64
```

[10]:
```
## Computing Mean Rating
restaurants = list(zomato_df['name'].unique())
zomato_df['Mean Rating'] = 0
for i in range(len(restaurants)):
    zomato_df['Mean Rating'][zomato_df['name'] == restaurants[i]] =␣
 ↪zomato_df['rate'][zomato_df['name'] == restaurants[i]].mean()
#Scaling the mean rating values
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (1,5))
zomato_df[['Mean Rating']] = scaler.fit_transform(zomato_df[['Mean Rating']]).
 ↪round(2)
```

[11]:
```
zomato_df[['name','rate','Mean Rating']].head()
```

[11]:

	name	rate	Mean Rating
0	Jalsa	4.1	3.99
1	Spice Elephant	4.1	3.97
2	San Churro Cafe	3.8	3.58
3	Addhuri Udupi Bhojana	3.7	3.45
4	Grand Village	3.8	3.58

[12]:
```
## Lower Casing
zomato_df["reviews_list"] = zomato_df["reviews_list"].str.lower()

## Removal of Puctuations
import string
PUNCT_TO_REMOVE = string.punctuation
def remove_punctuation(text):
    """custom function to remove the punctuation"""
    return text.translate(str.maketrans('', '', PUNCT_TO_REMOVE))
zomato_df["reviews_list"] = zomato_df["reviews_list"].apply(lambda text:␣
 ↪remove_punctuation(text))
```

[13]:
```
zomato_df[['reviews_list', 'cuisines','url']].sample(5)
```

[13]:

	reviews_list	cuisines	url
2565	rated 20 ratedn bad quality of puff and bomba…	Bakery	https://www.zomato.com/bangalore/cake-art-basa…
32714	rated 40 ratedn limited options in the menuth…	Continental, North Indian, Chinese, Arabian	https://www.zomato.com/bangalore/high-sky-whit…
12842	rated 30 ratedn rude behavior by the staff ve…	North Indian, Mughlai, Mediterranean, Iranian	https://www.zomato.com/bangalore/ruh-bellandur…
30607	rated 30 ratedn been there on several occasio…	North Indian, Mithai	https://www.zomato.com/bangalore/bhaiyaji-food…
29166	rated 10 ratedn service was very disappointin…	Chinese, Thai, Asian	https://www.zomato.com/bangalore/magnolia-kora…

[14]:
```
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]
```

[15]:
```
# RESTAURANT NAMES:
restaurant_names = list(zomato_df['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_df=zomato_df.drop(['address','rest_type', 'type', 'menu_item',␣
 ↪'votes'],axis=1)


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

[16]:
```
zomato_df.head()
```

[16]:

	url	name	online_order	book_table	rate	location	cuisines	cost	reviews_list	city	Mean Rating
0	https://www.zomato.com/bangalore/jalsa-banasha…	Jalsa	Yes	Yes	4.1	Banashankari	North Indian, Mughlai, Chinese	800.0	rated 40 ratedn a beautiful place to dine int…	Banashankari	3.99
1	https://www.zomato.com/bangalore/spice-elephan…	Spice Elephant	Yes	No	4.1	Banashankari	Chinese, North Indian, Thai	800.0	rated 40 ratedn had been here for dinner with…	Banashankari	3.97
2	https://www.zomato.com/SanchurroBangalore?cont…	San Churro Cafe	Yes	No	3.8	Banashankari	Cafe, Mexican, Italian	800.0	rated 30 ratedn ambience is not that good eno…	Banashankari	
3	https://www.zomato.com/bangalore/addhuri-udupi…	Addhuri Udupi Bhojana	No	No	3.7	Banashankari	South Indian, North Indian	300.0	rated 40 ratedn great food and proper karnata…	Banashankari	
4	https://www.zomato.com/bangalore/grand-village…	Grand Village	No	No	3.8	Basavanagudi	North Indian, Rajasthani	600.0	rated 40 ratedn very good restaurant in neigh…	Banashankari	

```
2         3.58
3         3.45
4         3.58
```

[17]:
```
zomato_df.to_csv("restaurant1.csv")
```

[18]:
```
zomato_df.to_csv("restaurant2.csv")
```

[19]:
```
df_percent.head()
```

[19]:
```
                                                    url  \
5676   https://www.zomato.com/bangalore/kfc-3-whitefi…
31391  https://www.zomato.com/bangalore/corner-house-…
5181   https://www.zomato.com/bangalore/just-bake-sha…
31359  https://www.zomato.com/bangalore/tea-samakruth…
11609  https://www.zomato.com/bangalore/taco-bell-ind…

                        name online_order book_table  rate  \
5676                     KFC          Yes         No   2.8
31391  Corner House Ice Cream          Yes         No   4.4
5181                Just Bake          Yes         No   3.5
31359         Tea Samskruthi           No         No   3.9
11609              Taco Bell          Yes         No   4.1

                        location                       cuisines   cost  \
5676   ITPL Main Road, Whitefield             Burger, Fast Food  400.0
31391               Seshadripuram           Ice Cream, Desserts  400.0
5181                 Shanti Nagar              Bakery, Desserts  400.0
31359                Malleshwaram                          Cafe  200.0
11609                 Indiranagar  Mexican, American, Fast Food  600.0

                                          reviews_list          city  \
5676   rated 40 ratedn  in banglore there are many kf…   Brookefield
31391  rated 40 ratedn  very close to my work place w…  Malleshwaram
5181   rated 50 ratedn  just bake cake is just awesom…  Brigade Road
31359  rated 50 ratedn  tea samskruti is the best pla…  Malleshwaram
11609  rated 40 ratedn  good place for mexican food t…   Frazer Town

       Mean Rating
5676          3.38
31391         4.44
5181          3.07
31359         3.71
11609         3.81
```

[20]:
```
df_percent['reviews_list'].isnull().sum()
```

[20]: 0

[21]:
```
df_percent['url'].isnull().sum()
```

[21]: 0

[22]:
```
df_percent.set_index('name', inplace=True)
indices = pd.Series(df_percent.index)

# Creating tf-idf matrix
tfidf = TfidfVectorizer(analyzer='word', ngram_range=(1, 2), min_df=0,␣
 ↪stop_words='english')
tfidf_matrix = tfidf.fit_transform(df_percent['reviews_list'])

cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
```

[23]:
```
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','url'])

    # 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','url']][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','url'], 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
```

[24]:
```
recommend('Pai Vihar')
```

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

[24]:

	cuisines	Mean Rating	cost
Cinnamon	North Indian, Asian, Continental	3.62	1.0
Samosa Singh	Street Food, Fast Food, Rolls, Desserts	3.60	200.0
Samosa Singh	Street Food, Beverages	3.60	150.0
Kadai Crust - Amma Veetu Samayal	Chettinad, South Indian, Biryani	3.58	700.0
Pallavi Restaurant	Biryani, Chinese, Andhra	3.58	500.0
Upahar Sagar	South Indian, Chinese, North Indian	3.58	350.0
Magix's Parattha Roll	Fast Food, North Indian, Chinese, Mughlai, Rolls	3.52	400.0
Magix's Parattha Roll	Fast Food, North Indian, Chinese, Mughlai, Rolls	3.52	400.0
Magix's Parattha Roll	Fast Food, North Indian, Chinese, Mughlai, Rolls	3.52	400.0
Prasiddhi Food Corner	Fast Food, North Indian, South Indian	3.45	200.0

```
                                                                  url
Cinnamon                                https://www.zomato.com/bangalore/cinnamon-
sesh…
Samosa Singh                            https://www.zomato.com/bangalore/samosa-
singh-…
Samosa Singh                            https://www.zomato.com/bangalore/samosa-
singh-…
Kadai Crust - Amma Veetu Samayal  https://www.zomato.com/bangalore/kadai-
crust-a…
Pallavi Restaurant                      https://www.zomato.com/bangalore/pallavi-
resta…
Upahar Sagar                            https://www.zomato.com/bangalore/upahar-
sagar-…
Magix's Parattha Roll                   https://www.zomato.com/bangalore/magixs-
paratt…
Magix's Parattha Roll                   https://www.zomato.com/bangalore/magixs-
paratt…
Magix's Parattha Roll                   https://www.zomato.com/bangalore/magixs-
paratt…
Prasiddhi Food Corner                   https://www.zomato.com/bangalore/prasiddhi-
foo…
```

[25]:
```
recommend('Canopy')
```

```
TOP 10 RESTAURANTS LIKE Canopy WITH SIMILAR REVIEWS:
```

[25]:
```
                                                cuisines  \
Atithi                                    North Indian, Chinese, Street
Food
Atithi                                    North Indian, Chinese, Street
Food
Cinnamon                                      North Indian, Chinese,
Biryani
Cafe @ Elanza                                    Chinese, North Indian,
Cafe
Cafe @ Elanza                                    Chinese, North Indian,
Cafe
Nouvelle Garden                           North Indian, Continental,
Italian
Sri Sai Mango Tree Restaurant                 North Indian, Biryani,
Chinese
The Onyx - The HHI Select Bengaluru       North Indian, Chinese,
Continental
Wazir's                                                   North Indian,
Chinese
Melange - Hotel Ekaa            North Indian, Chinese, Continental,
Mangalorean
```

```
                                    Mean Rating   cost  \
Atithi                                     3.63  800.0
Atithi                                     3.63  800.0
Cinnamon                                   3.62  550.0
Cafe @ Elanza                              3.45    1.0
Cafe @ Elanza                              3.45    1.0
Nouvelle Garden                            3.45  900.0
Sri Sai Mango Tree Restaurant              3.32  600.0
The Onyx - The HHI Select Bengaluru        2.97  950.0
Wazir's                                    2.94  500.0
Melange - Hotel Ekaa                       2.81  900.0

                                                   url
Atithi                                 https://www.zomato.com/bangalore/atithi-
hsr?co…
Atithi                                 https://www.zomato.com/bangalore/atithi-
hsr?co…
Cinnamon                               https://www.zomato.com/bangalore/cinnamon-
hsr?…
Cafe @ Elanza                          https://www.zomato.com/bangalore/cafe-
elanza-r…
Cafe @ Elanza                          https://www.zomato.com/bangalore/cafe-
elanza-r…
Nouvelle Garden                        https://www.zomato.com/bangalore/nouvelle-
gard…
Sri Sai Mango Tree Restaurant          https://www.zomato.com/bangalore/sri-sai-
mango…
The Onyx - The HHI Select Bengaluru    https://www.zomato.com/bangalore/the-onyx-
the-…
Wazir's                                https://www.zomato.com/bangalore/wazirs-
shanti…
Melange - Hotel Ekaa                   https://www.zomato.com/bangalore/melange-
hotel…
```

[26]:
```
recommend('Cinnamon')
```

```
TOP 10 RESTAURANTS LIKE Cinnamon WITH SIMILAR REVIEWS:
```

[26]:
```
cuisines  \
Chianti
Italian
Chianti
Italian
Chinita Real Mexican Food
Mexican
Oh! Calcutta                                                    Bengali,
```

```
Seafood
Oh! Calcutta                                                    Bengali,
Seafood
Soda Bottle Opener Wala                                      Parsi, North
Indian
CafÃ Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â …          American, Cafe,
Continental
CafÃ Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â …          American, Cafe,
Continental
CafÃ Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â …          American, Cafe,
Continental
Foxtrot - House of Subculture                        Cafe, American, Asian, North
Indian

                                                    Mean Rating  cost  \
Chianti                                                    4.59   1.5
Chianti                                                    4.59   1.5
Chinita Real Mexican Food                                  4.47   1.2
Oh! Calcutta                                               4.39   1.2
Oh! Calcutta                                               4.39   1.2
Soda Bottle Opener Wala                                    4.36   1.3
CafÃ Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â …        4.35   1.7
CafÃ Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â …        4.35   1.7
CafÃ Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â …        4.35   1.7
Foxtrot - House of Subculture                              4.35   1.0

                  url
Chianti
https://www.zomato.com/bangalore/chianti-koram…
Chianti
https://www.zomato.com/bangalore/chianti-mg-ro…
Chinita Real Mexican Food
https://www.zomato.com/bangalore/chinita-real-…
Oh! Calcutta
https://www.zomato.com/bangalore/oh-calcutta-c…
Oh! Calcutta
https://www.zomato.com/bangalore/oh-calcutta-c…
Soda Bottle Opener Wala
https://www.zomato.com/bangalore/soda-bottle-o…
CafÃ Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â …
https://www.zomato.com/bangalore/caf%C3%A9-fel…
CafÃ Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â …
https://www.zomato.com/bangalore/caf%C3%A9-fel…
CafÃ Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â Ã Â …
https://www.zomato.com/bangalore/caf%C3%A9-fel…
Foxtrot - House of Subculture
https://www.zomato.com/bangalore/foxtrot-house…
```

[27]:
```
import pickle
pickle.dump(tfidf, open('restaurant2.pkl', 'wb'))
```

[ ]:
```

```