RESTAURANT RECOMMENDATION SYSTEM
Team Id - 592691

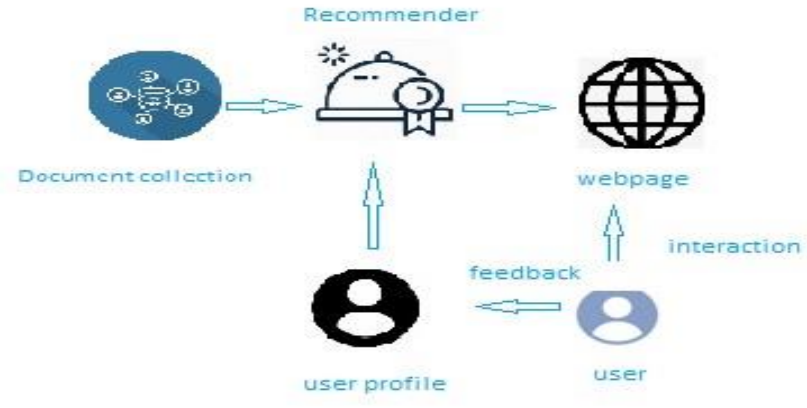
PROJECT DESCRIPTION

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.

SOLUTION

Here we are creating a content-based recommendation system. The aim is to create a content-based recommender system in which when we will write a restaurant name, the Recommender system will **look at the reviews of other restaurants**, and the System will recommend us other restaurants with **similar reviews** and sort them from the **highest-rated**. The main people who are going to benefit from this recommendation system are the tourists, who are new to a city. Most of the tourists always love to visit famous restaurants in a particular city during their visit. Otherwise, it can be heavily used by people belonging to the same city, to see if any new restaurant is recommended based on their activity.

ARCHITECTURE



LEARNING OUTCOMES

By the end of this project:

- You'll be able to perform one of the techniques to build you recommendation system
- You'll be able to know the recommendation system using Content-Based Filtering.
- You will be able to know how to pre-process / clean the data using different data pre-processing techniques.
- You will able to analyse or get insights of data through visualization.
- Applying algorithms according to dataset and based on visualization.
- You will able to know how to find accuracy of the model.
- You will be able to know how to build a web application using Flask framework.

PRE-REQUISITES:

To complete the project successfully, you need to install following software & packages:

Activity 1: Install Anaconda IDE / Anaconda Navigator.

- In order develop a solution to this problem statement, we need an environment to write and test the code.
- We use Anaconda IDE (Integrated Developing Environment).
- Refer to the below link to download & install Anaconda Navigator.

Link: https://www.youtube.com/watch?v=5mDYijMfSzs

1. Activity 2: To build Machine learning models you must require the following packages

- **Numpy**:

  It is an open-source numerical Python library. It contains a multidimensional array and matrix data structures and can be used to perform mathematical operations

- **Numpy**:

It is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy

- **Matplotlib and Seaborn**

  Matplotlib is mainly deployed for basic plotting. Visualization using Matplotlib generally consists of bars, pies, lines, scatter plots and so on. Seaborn: Seaborn, on the other hand, provides a variety of visualization patterns. It uses fewer syntax and has easily interesting default themes.

- **Flask:**

  Web framework used for building Web applications

If you are using **anaconda navigator**, follow below steps to download required packages:

- Open anaconda prompt.
- Type “pip install pandas” and click enter.
- Type “pip install matplotlib” and click enter.
- Type “pip install seaborn” and click enter.
- Type “pip install plotly” and click enter.
- Type “pip install numpy” and click enter.
- Type “pip install scikit-image” and click enter.
- Type “pip install scikit-learn” and click enter.
- Type “pip install Flask” and click enter.

Link: Introduction to Scikit-Learn (sklearn) in Python • datagy

PRIOR KNOWLEDGE

One should have knowledge on the following Concepts:
Link: Supervised and Unsupervised Learning

Watch the below video to know about the types of machine learning

Link: Regression, Classification and Clustering

Link: ML - Content Based Recommender System - GeeksforGeeks

Link: NLTK :: Natural Language Toolkit

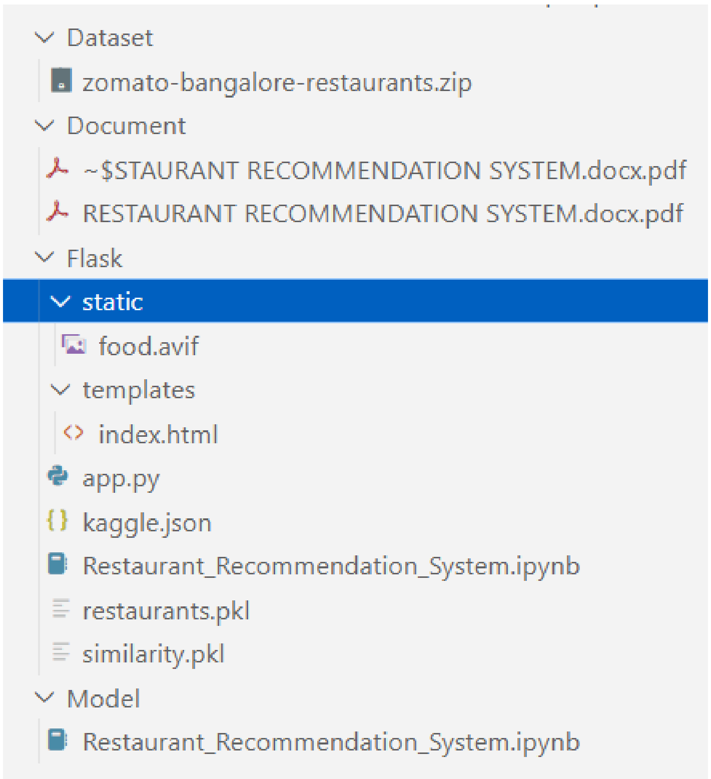
Link: Flask:

Link: Recommendation System

It is recommended to watch above video's to understand the concepts before you start your project.

PROJECT WORK FLOW

- User interacts with the UI (User Interface) to enter the input features.
- Entered features/input is analysed by the model which is integrated
- Once model analyses the entered inputs, the prediction is showcased on the UI.



```
Dataset
  zomato-bangalore-restaurants.zip
Document
  ~$STAURANT RECOMMENDATION SYSTEM.docx.pdf
  RESTAURANT RECOMMENDATION SYSTEM.docx.pdf
Flask
  static
    food.avif
  templates
    index.html
  app.py
  kaggle.json
  Restaurant_Recommendation_System.ipynb
  restaurants.pkl
  similarity.pkl
Model
  Restaurant_Recommendation_System.ipynb
```

TASKS

1. Data Collection.
   - Collect the dataset or Create the dataset

2. Data Pre- processing.
   - Import the Libraries.
   - Importing the dataset.
   - Exploratory Data Analysis
   - Data Visualization.

3. Content Based Filtering
   - Merging datasets
   - Creating the recommender system
   - Predicting the results

4. Application Building Create an HTML file
   - Build a Python Code

Milestone 1: Data Collection

Now, the milestone-2 is all about creation or collection of dataset.

we will use the **Zomato** Bangalore for our analysis to draw conclusions using the content filtering method.

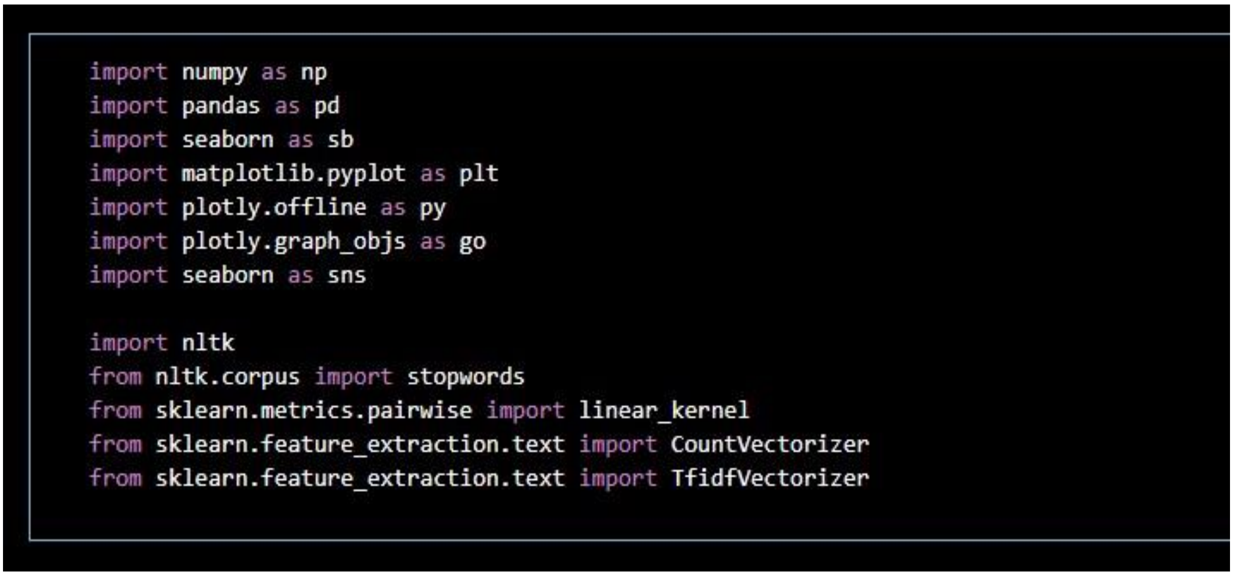
Here is the dataset link: **dataset**

Milestone 2: Data Pre-processing

In this milestone, you need to complete all the below activities to build the model.

Activity 1: Import Libraries

Import the below essential libraries for data pre-processing and creating recommendation system. Pandas and NumPy are used for data pre-processing and cleaning. Seaborn, Plotly and Matplotlib helped in creating visual graphics and bar plots for the dataset. Also, since there would be cleaning of text data (reviews) as well, therefore for that we will use nltk and sklearn library.

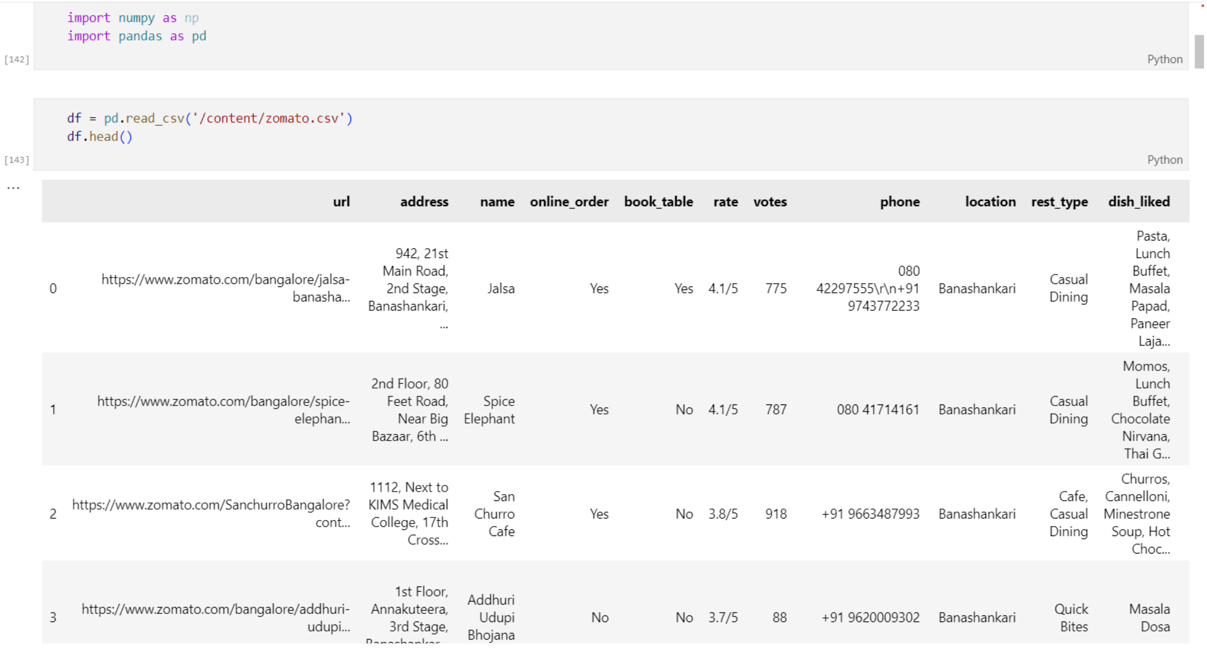


```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
import plotly.offline as py
import plotly.graph_objs as go
import seaborn as sns

import 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
```

Activity 2: Read the Dataset:

Our dataset format might be in .csv, excel files, .txt, json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read_csv () to read the dataset. As a parameter we have to give the directory of csv file.



```
import numpy as np
import pandas as pd
```

[142] Python

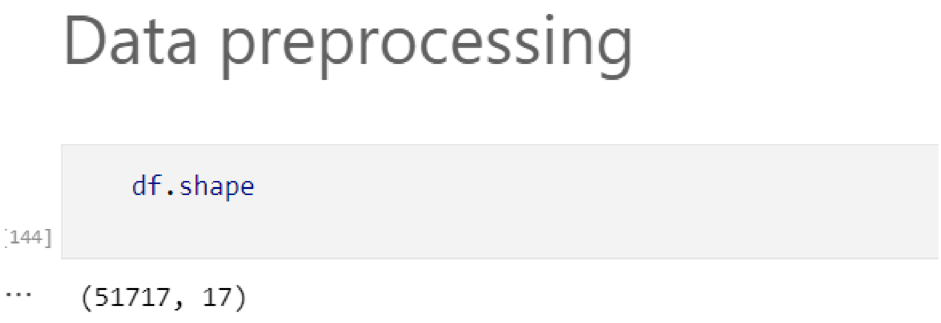
```
df = pd.read_csv('/content/zomato.csv')
df.head()
```

[143] Python

...

	url	address	name	online_order	book_table	rate	votes	phone	location	rest_type	dish_liked
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	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...
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	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...
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	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...
3	https://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage,	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 9620009302	Banashankari	Quick Bites	Masala Dosa

Activity 3: Analyse the Dataset:



Data preprocessing

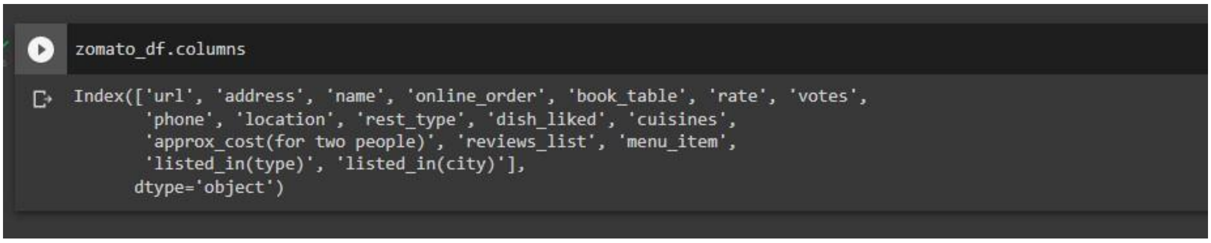
```
df.shape
```

```
144]
```

```
...    (51717, 17)
```

The dataset contains 51717 records with 17 features.

Checking the columns in the dataset.



```
zomato_df.columns
```

```
Index(['url', 'address', 'name', 'online_order', 'book_table', 'rate', 'votes',
       'phone', 'location', 'rest_type', 'dish_liked', 'cuisines',
       'approx_cost(for two people)', 'reviews_list', 'menu_item',
       'listed_in(type)', 'listed_in(city)'],
      dtype='object')
```

Columns description

1. **URL** contains the url of the restaurant on the Zomato website
2. **address** contains the address of the restaurant in Bengaluru
3. **name** contains the name of the restaurant
4. **online_order** whether online ordering is available in the restaurant or not
5. **book_table** table book option available or not
6. **rate** contains the overall rating of the restaurant out of 5
7. **votes** contain the total number of rating for the restaurant as of the above-mentioned date
8. **phone** contains the phone number of the restaurant

9. **location** contains the neighbourhood in which the restaurant is located

10. **rest_type** restaurant type

11. **dish_liked** dishes people liked in the restaurant

12. **cuisines** food styles, separated by comma

13. **approx_cost(for two people)** contains the approximate cost for a meal for two people

14. **reviews_list** list of tuples containing reviews for the restaurant, each tuple

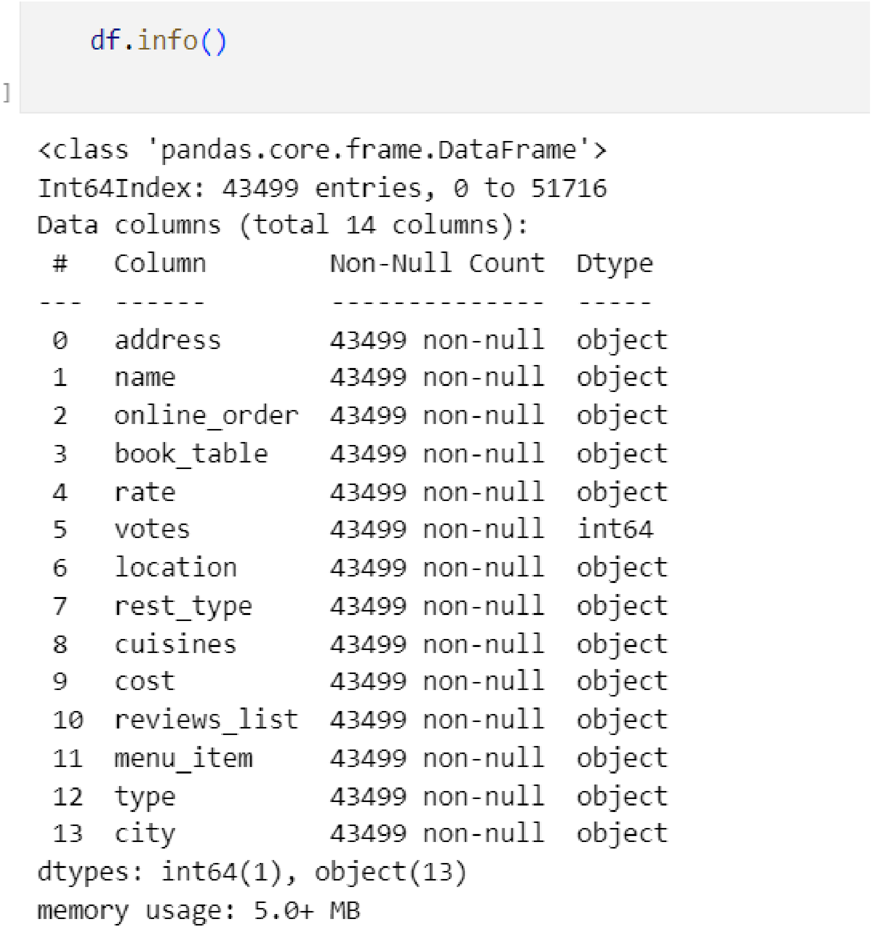
15. **menu_item** contains a list of menus available in the restaurant

16. **listed_in(type)** type of meal

17. **listed_in(city)** contains the neighbourhood in which the restaurant is listed.

Understanding Overview of features

- How the information is stored in a DataFrame or Python object affects what we can do with it and the outputs of calculations as well. There are two main types of data those are numeric and text data types.
- Numeric data types include integers and floats.
- Text data type is known as Strings in Python, or Objects in Pandas. Strings can contain numbers and / or characters.
- For example, a string might be a word, a sentence, or several sentences.
- Will see how our dataset is, by using **info ()** method.

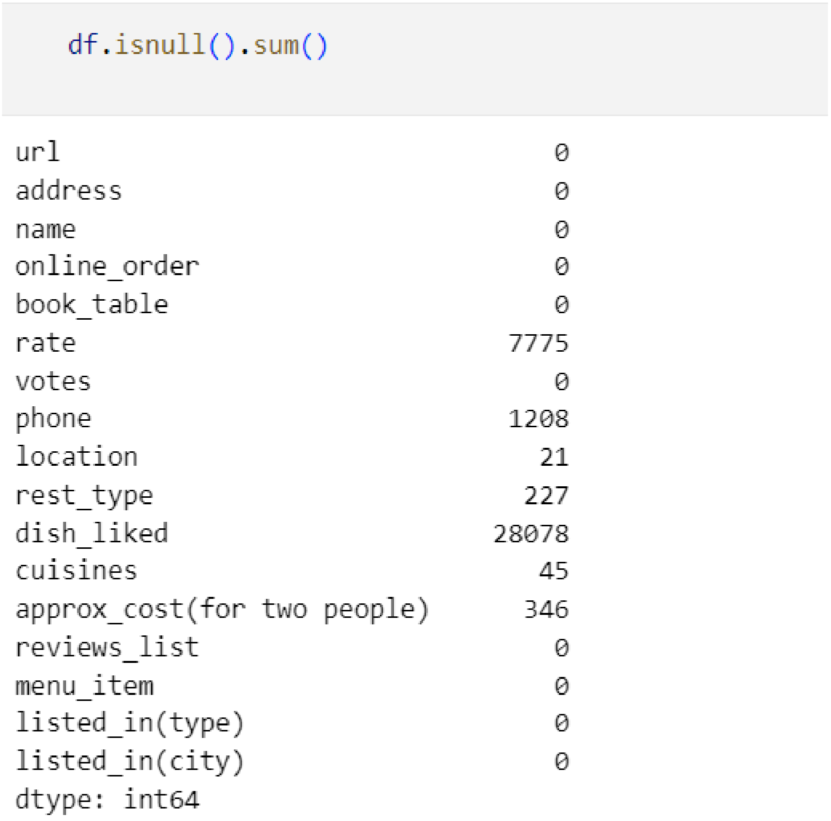


```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43499 entries, 0 to 51716
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   address       43499 non-null  object
 1   name          43499 non-null  object
 2   online_order  43499 non-null  object
 3   book_table    43499 non-null  object
 4   rate          43499 non-null  object
 5   votes         43499 non-null  int64 
 6   location      43499 non-null  object
 7   rest_type     43499 non-null  object
 8   cuisines      43499 non-null  object
 9   cost          43499 non-null  object
 10  reviews_list  43499 non-null  object
 11  menu_item     43499 non-null  object
 12  type          43499 non-null  object
 13  city          43499 non-null  object
dtypes: int64(1), object(13)
memory usage: 5.0+ MB
```

- As you can see in our dataset, except ‘votes’, all other features are categorical data, but it is not necessary that all the continuous data which we are seeing has to be continuous in nature. There may be a case that some categorical data is in the form of numbers but when we perform info () operation we will get numerical output. So, we need to take care of those type of data also.

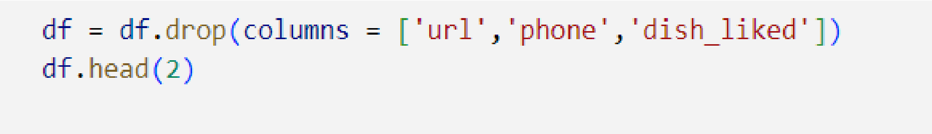
Checking for null values in the dataset



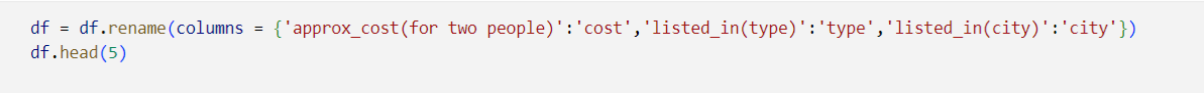
```
df.isnull().sum()
```

```
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
```

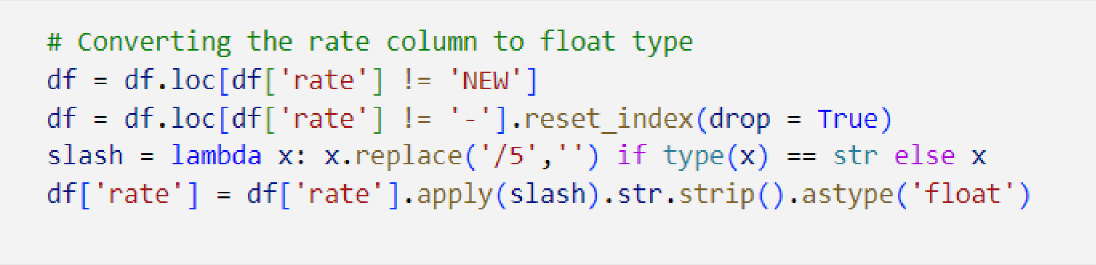
Data cleaning as our dataset contains null values and some special characters



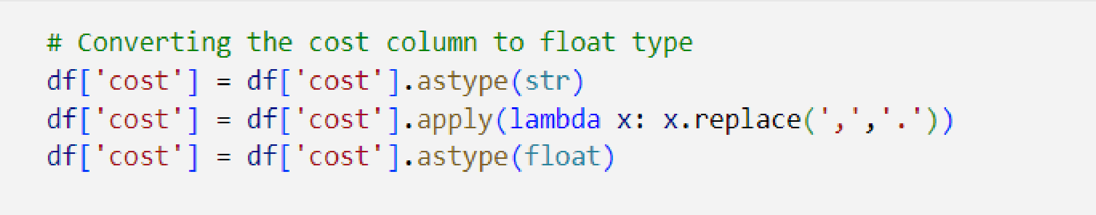
```
df = df.drop(columns = ['url','phone','dish_liked'])
df.head(2)
```



```
df = df.rename(columns = {'approx_cost(for two people)':'cost','listed_in(type)':'type','listed_in(city)':'city'})
df.head(5)
```

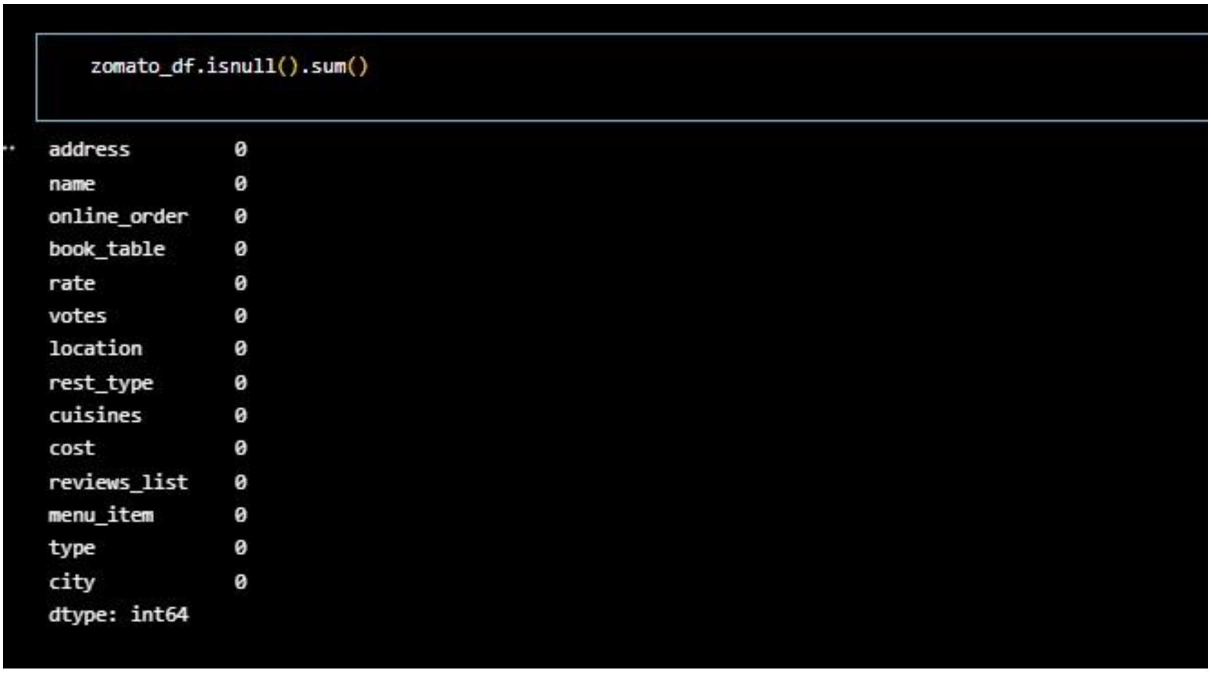


```
# Converting the rate column to float type
df = df.loc[df['rate'] != 'NEW']
df = df.loc[df['rate'] != '-'].reset_index(drop = True)
slash = lambda x: x.replace('/5','') if type(x) == str else x
df['rate'] = df['rate'].apply(slash).str.strip().astype('float')
```



```
# Converting the cost column to float type
df['cost'] = df['cost'].astype(str)
df['cost'] = df['cost'].apply(lambda x: x.replace(',','.'))
df['cost'] = df['cost'].astype(float)
```

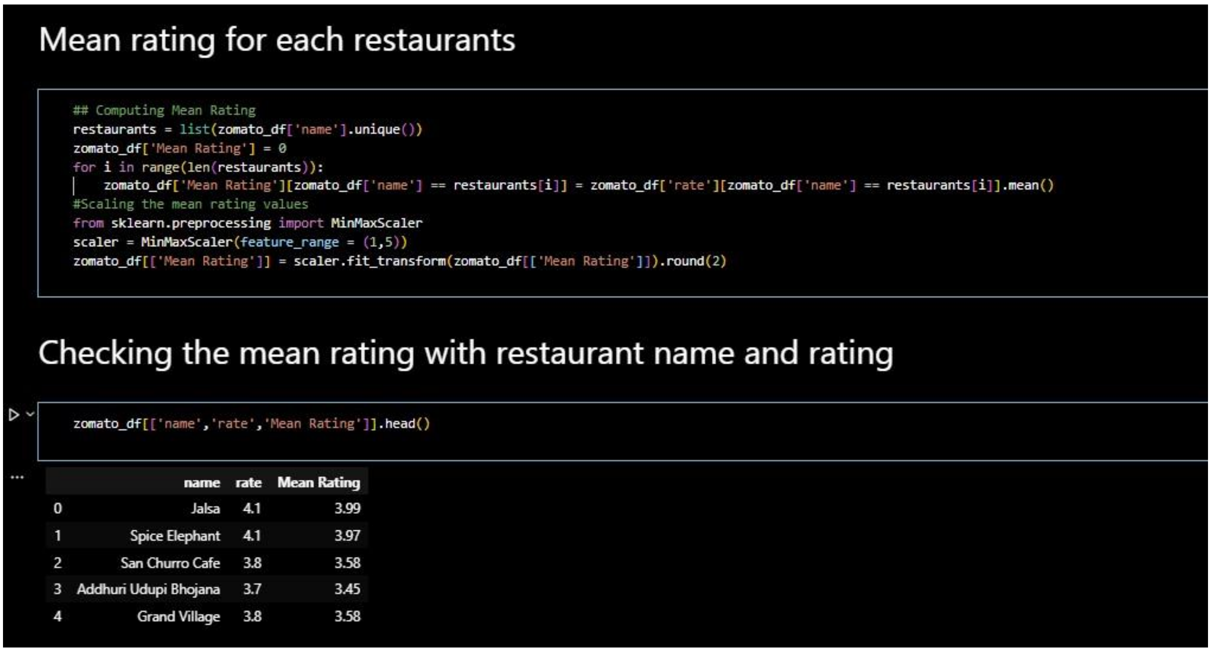
Checking for null values after cleaning & data Processing



```
zomato_df.isnull().sum()
```

```
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
```

Checking mean rating with restaurant name and rating for each restaurant using below line codes



Mean rating for each restaurants

```
## 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)
```

Checking the mean rating with restaurant name and rating

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

```
                    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
```

We will be using the 'Review' and 'Cuisines' feature in order to create a recommender system. So we need to prepare and clean the text in those columns.

Operations performed: Lower Casing, Removal of Punctuations, Removal of Stop words, Removal of URLs, Spelling correction

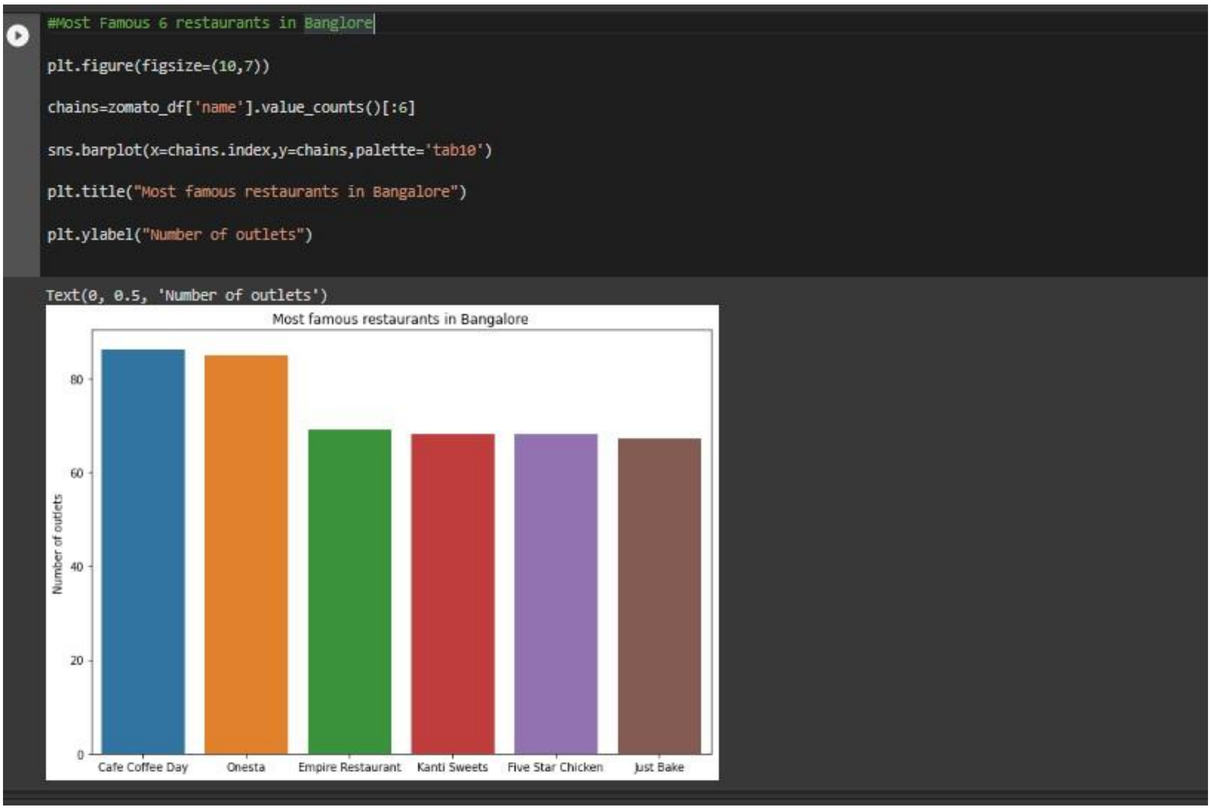
Milestone 3: Data Visualization

Data visualization is where a given data set is presented in a graphical format. It helps the detection of patterns, trends and correlations that might go undetected in text-based data. Understanding your data and the relationship present within it is just as important as any algorithm used to train your machine learning model. In fact, even the most sophisticated machine learning models will perform poorly on data that wasn't visualized and understood properly.

To visualize the dataset, we need libraries called Matplotlib, Seaborn. The Matplotlib library is a Python 2D plotting library which allows you to generate plots, scatter plots, histograms, bar charts etc.

Let's visualize our data using Matplotlib and seaborn library.

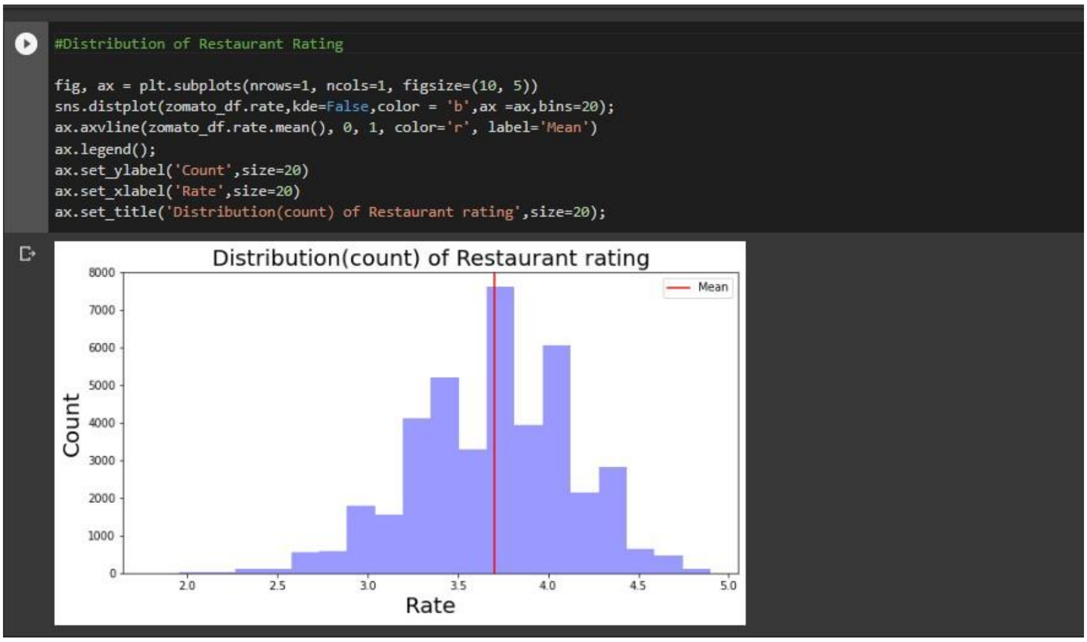
At first, we will be plotting a bar plot using matplotlib for showing the top 6 restaurants in Bangalore by value counts.



```
#Most Famous 6 restaurants in Banglore

plt.figure(figsize=(10,7))

chains=zomato_df['name'].value_counts()[:6]

sns.barplot(x=chains.index,y=chains,palette='tab10')

plt.title("Most famous restaurants in Bangalore")

plt.ylabel("Number of outlets")
```

```
Text(0, 0.5, 'Number of outlets')
```

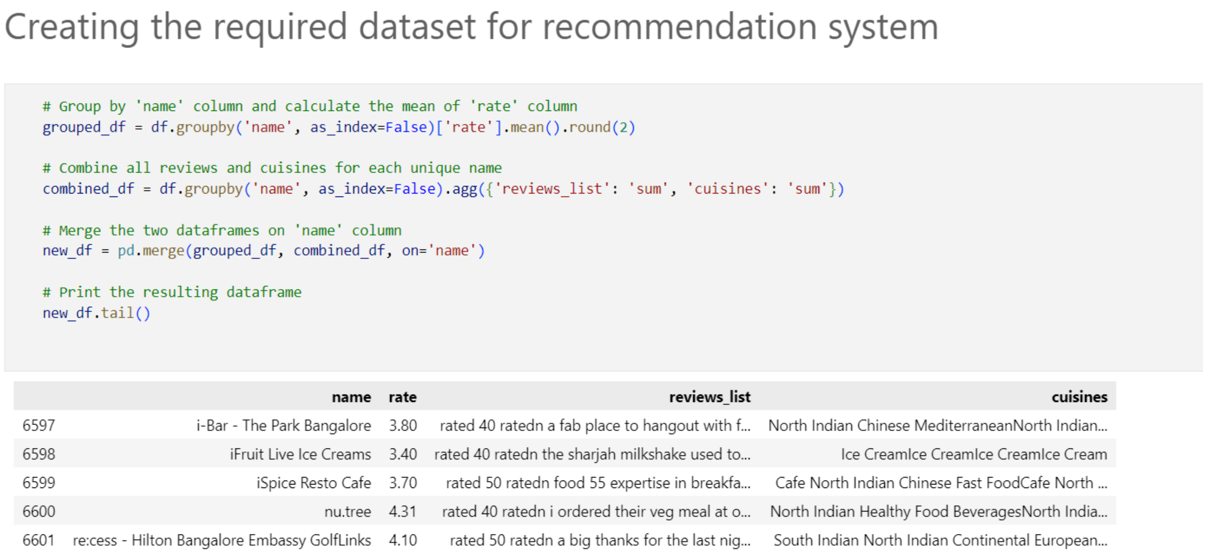
Checking the distribution of restaurant rating, for that we are using distplot from seaborn library.



```
#Distribution of Restaurant Rating

fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(10, 5))
sns.distplot(zomato_df.rate,kde=False,color = 'b',ax =ax,bins=20);
ax.axvline(zomato_df.rate.mean(), 0, 1, color='r', label='Mean')
ax.legend();
ax.set_ylabel('Count',size=20)
ax.set_xlabel('Rate',size=20)
ax.set_title('Distribution(count) of Restaurant rating',size=20);
```

Here we can see the Top favourite cuisine among people of Bangalore is ‘North Indian’, ‘Indian Chinese’ and ‘Fast food’.

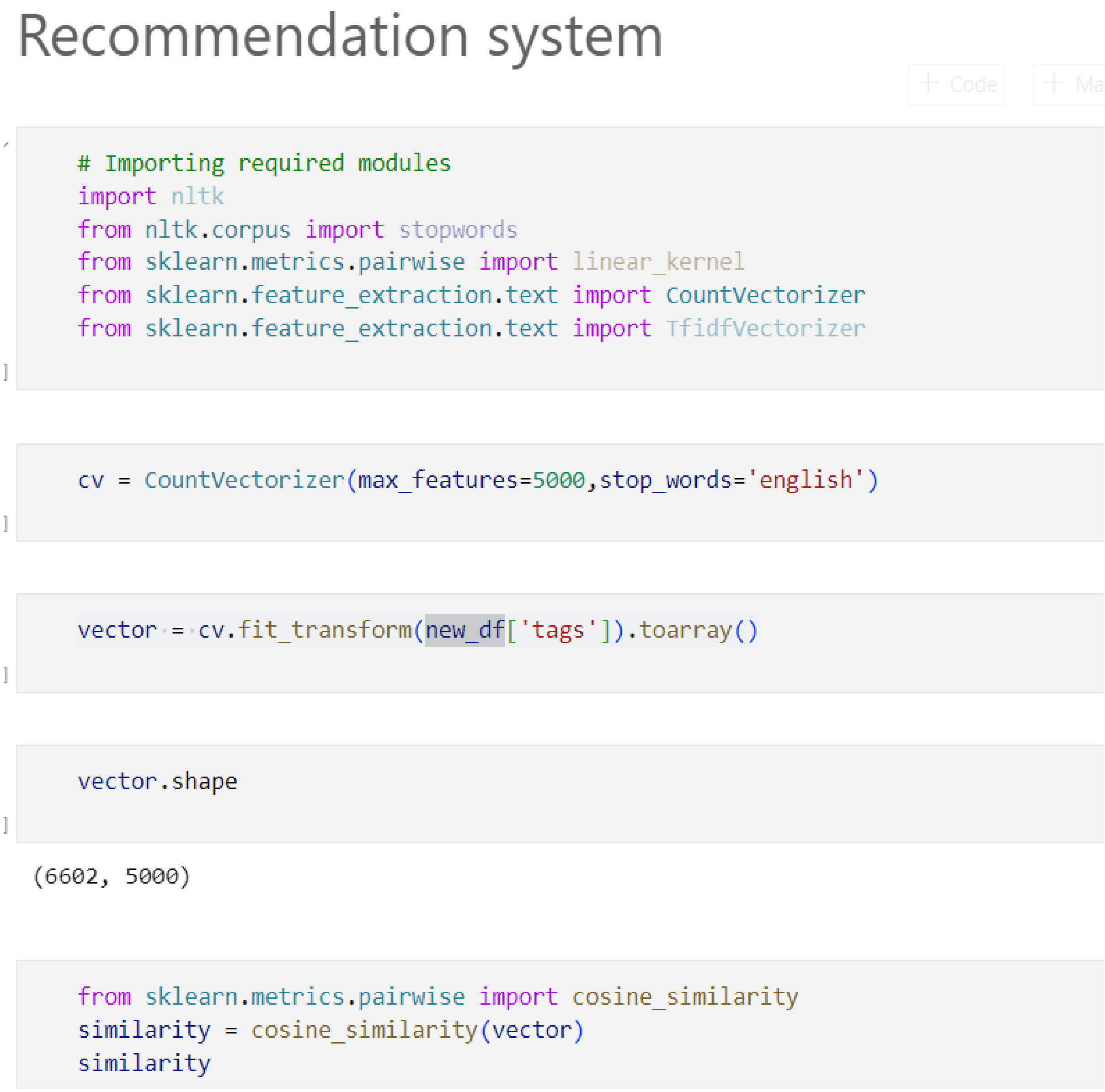
Milestone 4: CONTENT-BASE RECOMMENDER SYSTEM



Creating the required dataset for recommendation system

```
# Group by 'name' column and calculate the mean of 'rate' column
grouped_df = df.groupby('name', as_index=False)['rate'].mean().round(2)

# Combine all reviews and cuisines for each unique name
combined_df = df.groupby('name', as_index=False).agg({'reviews_list': 'sum', 'cuisines': 'sum'})

# Merge the two dataframes on 'name' column
new_df = pd.merge(grouped_df, combined_df, on='name')

# Print the resulting dataframe
new_df.tail()
```

	name	rate	reviews_list	cuisines
6597	i-Bar - The Park Bangalore	3.80	rated 40 ratedn a fab place to hangout with f...	North Indian Chinese MediterraneanNorth Indian...
6598	iFruit Live Ice Creams	3.40	rated 40 ratedn the sharjah milkshake used to...	Ice CreamIce CreamIce CreamIce Cream
6599	iSpice Resto Cafe	3.70	rated 50 ratedn food 55 expertise in breakfa...	Cafe North Indian Chinese Fast FoodCafe North ...
6600	nu.tree	4.31	rated 40 ratedn i ordered their veg meal at o...	North Indian Healthy Food BeveragesNorth India...
6601	re:cess - Hilton Bangalore Embassy GolfLinks	4.10	rated 50 ratedn a big thanks for the last nig...	South Indian North Indian Continental European...



Recommendation system

```
# Importing required modules
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
```

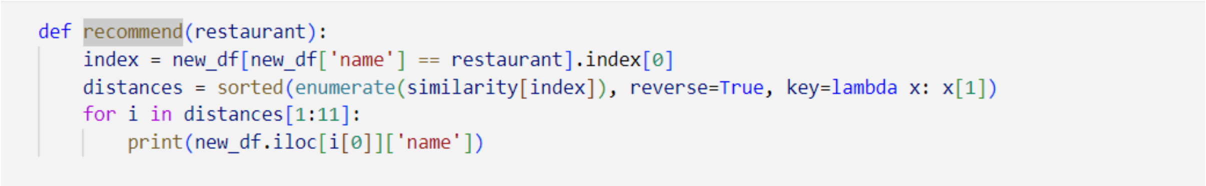
```
cv = CountVectorizer(max_features=5000,stop_words='english')
```

```
vector = cv.fit_transform(new_df['tags']).toarray()
```

```
vector.shape
```

```
(6602, 5000)
```

```
from sklearn.metrics.pairwise import cosine_similarity
similarity = cosine_similarity(vector)
similarity
```



```
def recommend(restaurant):
    index = new_df[new_df['name'] == restaurant].index[0]
    distances = sorted(enumerate(similarity[index]), reverse=True, key=lambda x: x[1])
    for i in distances[1:11]:
        print(new_df.iloc[i[0]]['name'])
```

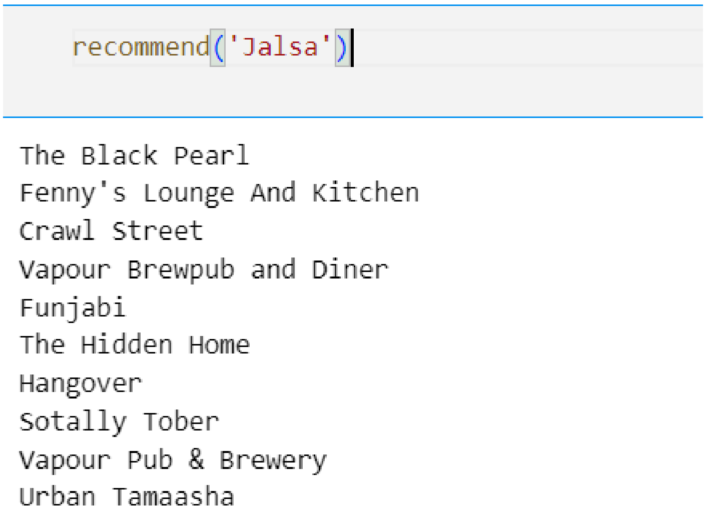
Calculating Cosine Similarity

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

The formula for Cosine Similarity

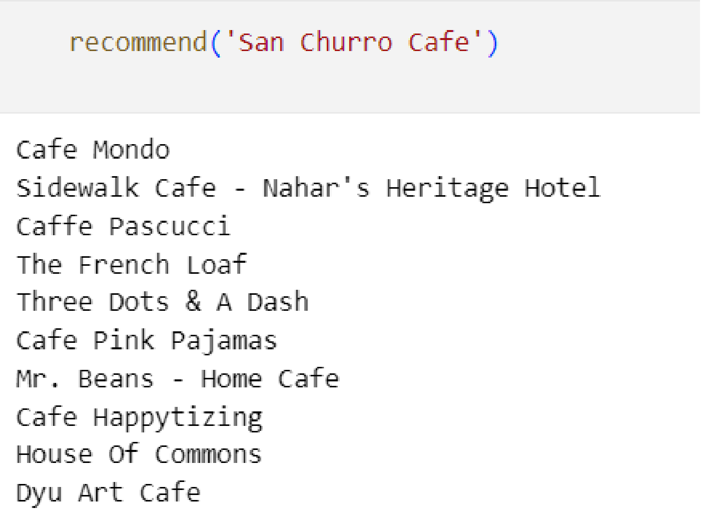
And in the last line of code, we are calculating the cosine similarity of each item with every other item in the dataset. So we just pass the matrix as an argument.

Querying recommendation for 4 Restaurants:



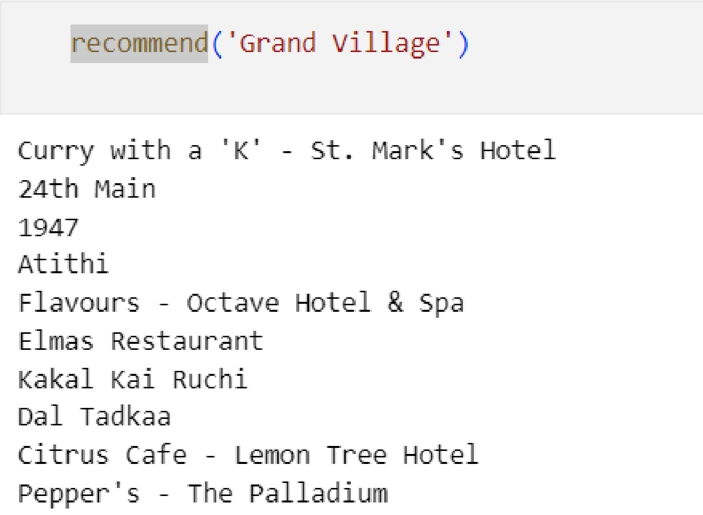
```
recommend('Jalsa')
```

```
The Black Pearl
Fenny's Lounge And Kitchen
Crawl Street
Vapour Brewpub and Diner
Funjabi
The Hidden Home
Hangover
Sotally Tober
Vapour Pub & Brewery
Urban Tamaasha
```



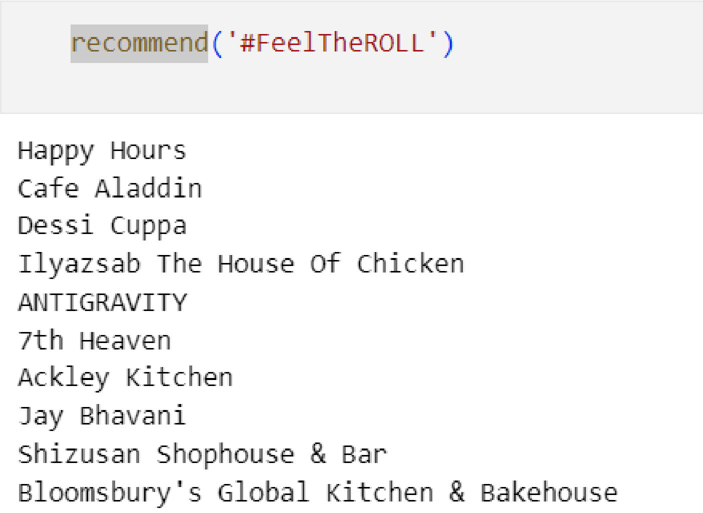
```
recommend('San Churro Cafe')
```

```
Cafe Mondo
Sidewalk Cafe - Nahar's Heritage Hotel
Caffe Pascucci
The French Loaf
Three Dots & A Dash
Cafe Pink Pajamas
Mr. Beans - Home Cafe
Cafe Happytizing
House Of Commons
Dyu Art Cafe
```



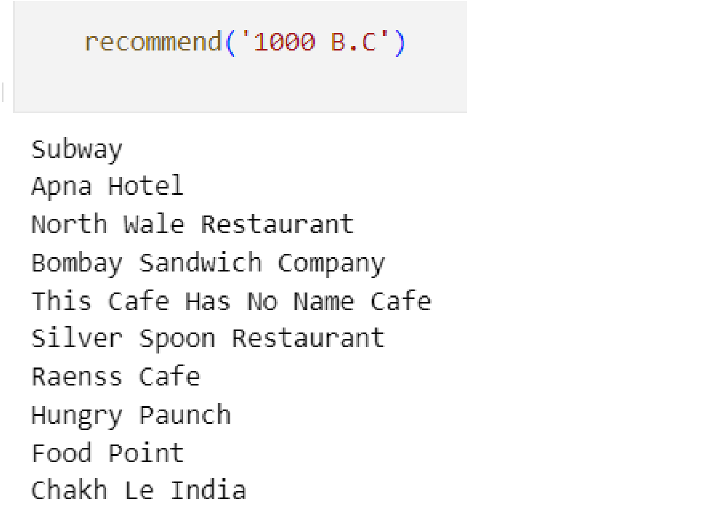
```
recommend('Grand Village')
```

```
Curry with a 'K' - St. Mark's Hotel
24th Main
1947
Atithi
Flavours - Octave Hotel & Spa
Elmas Restaurant
Kakal Kai Ruchi
Dal Tadkaa
Citrus Cafe - Lemon Tree Hotel
Pepper's - The Palladium
```



```
recommend('#FeelTheROLL')
```

```
Happy Hours
Cafe Aladdin
Dessi Cuppa
Ilyazsab The House Of Chicken
ANTIGRAVITY
7th Heaven
Ackley Kitchen
Jay Bhavani
Shizusan Shophouse & Bar
Bloomsbury's Global Kitchen & Bakehouse
```



```
recommend('1000 B.C')
```

```
Subway
Apna Hotel
North Wale Restaurant
Bombay Sandwich Company
This Cafe Has No Name Cafe
Silver Spoon Restaurant
Raenss Cafe
Hungry Paunch
Food Point
Chakh Le India
```

Milestone 5: Application Building

Activity 1: Create an HTML File

We use HTML to create the front end part of the web page.

Here, we created 2 html pages- index.html, web.html. index.html displays home page. web.html accepts the values from the input and displays the prediction.

For more information regarding HTML refer the link below

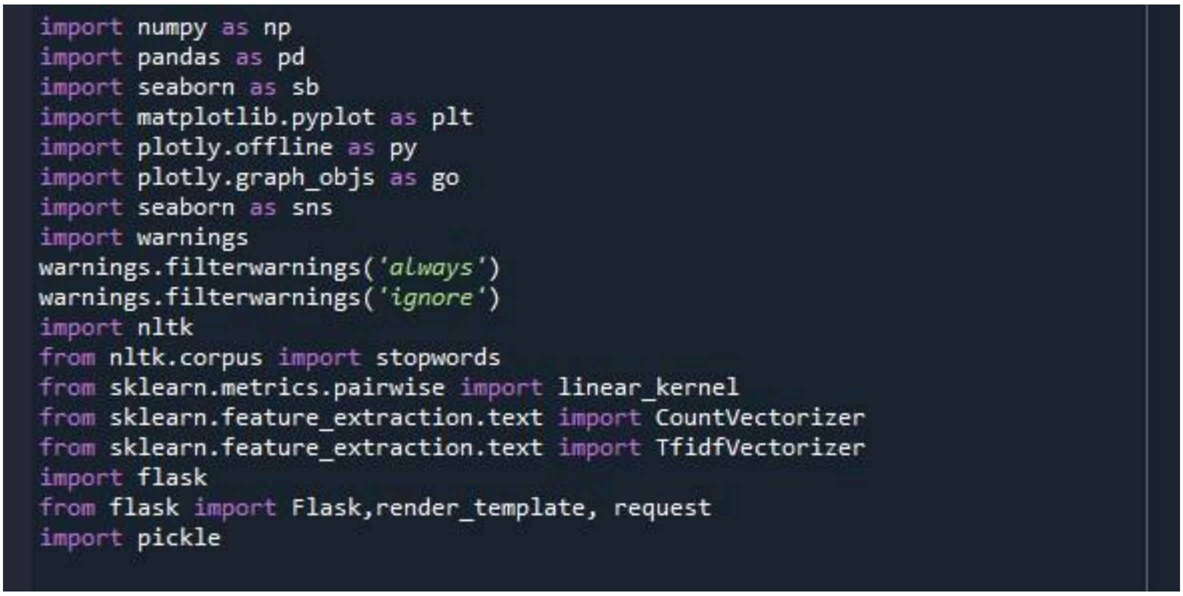
https://www.w3schools.com/bootstrap/bootstrap_forms_inputs.asp

- We also use JavaScript-main.js and CSS-main.css to enhance our functionality and view of HTML pages.
  - **Link :https://www.w3schools.com/css/**
  - **https://www.w3schools.com/js/DEFAULT.asp**

Activity 2: Build python code

- Let us build flask file ‘app1.py’ which is a web framework written in python for server-side scripting. Let’s see step by step procedure for building the backend application.
- App starts running when “__name__” constructor is called in main.
- render template is used to return html file.
- “GET” method is used to take input from the user.
- “POST” method is used to display the output to the user.

Importing libraries

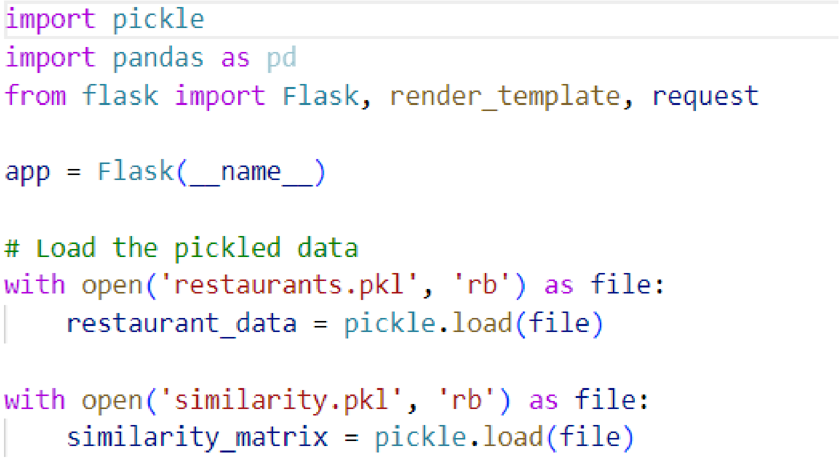


```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
import plotly.offline as py
import plotly.graph_objs as go
import seaborn as sns
import warnings
warnings.filterwarnings('always')
warnings.filterwarnings('ignore')
import nltk
from nltk.corpus import stopwords
from sklearn.metrics.pairwise import linear_kernel
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
import flask
from flask import Flask,render_template, request
import pickle
```

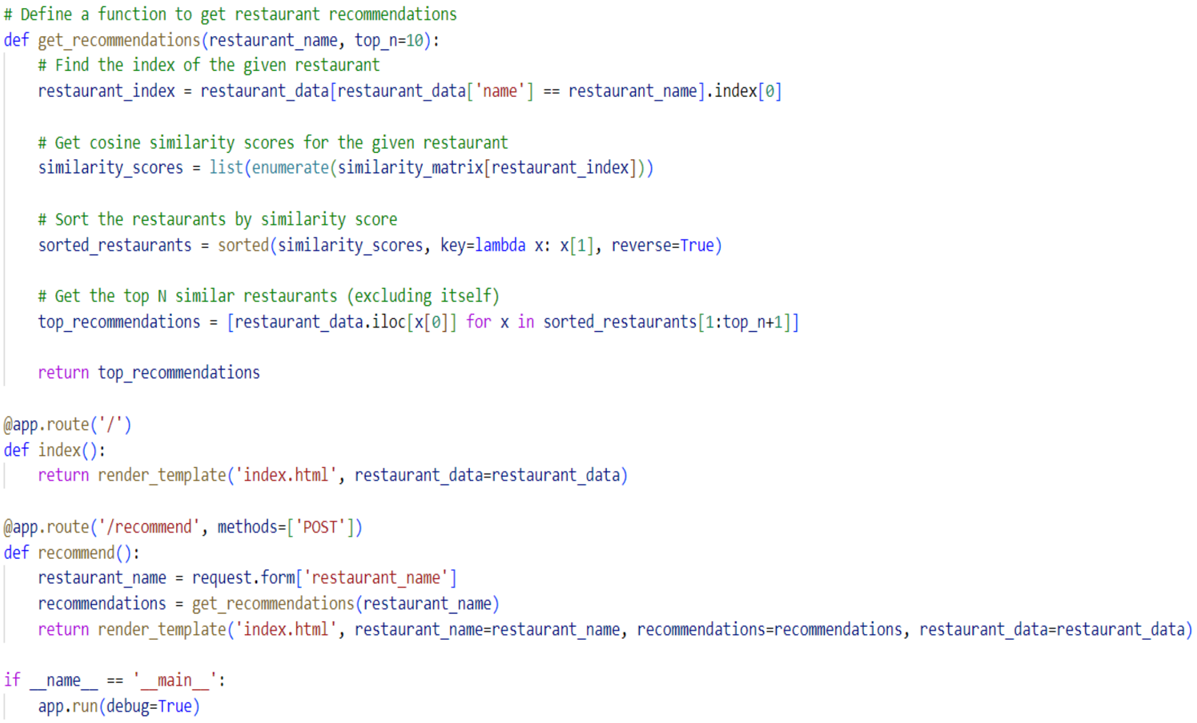
Libraries required for the app to run are to be imported.

Creating our flask app and loading the newly created dataset

Now after all the libraries are import we will be creating our flask app with the updated dataset

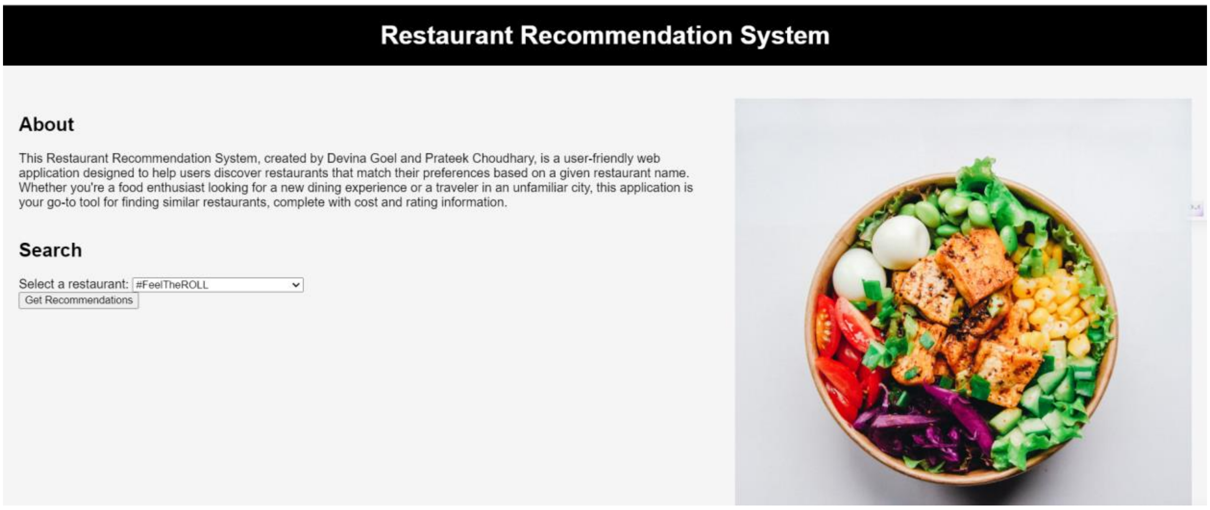


```python
import pickle
import pandas as pd
from flask import Flask, render_template, request

app = Flask(__name__)

# Load the pickled data
with open('restaurants.pkl', 'rb') as file:
    restaurant_data = pickle.load(file)

with open('similarity.pkl', 'rb') as file:
    similarity_matrix = pickle.load(file)
```



```python
# Define a function to get restaurant recommendations
def get_recommendations(restaurant_name, top_n=10):
    # Find the index of the given restaurant
    restaurant_index = restaurant_data[restaurant_data['name'] == restaurant_name].index[0]

    # Get cosine similarity scores for the given restaurant
    similarity_scores = list(enumerate(similarity_matrix[restaurant_index]))

    # Sort the restaurants by similarity score
    sorted_restaurants = sorted(similarity_scores, key=lambda x: x[1], reverse=True)

    # Get the top N similar restaurants (excluding itself)
    top_recommendations = [restaurant_data.iloc[x[0]] for x in sorted_restaurants[1:top_n+1]]

    return top_recommendations

@app.route('/')
def index():
    return render_template('index.html', restaurant_data=restaurant_data)

@app.route('/recommend', methods=['POST'])
def recommend():
    restaurant_name = request.form['restaurant_name']
    recommendations = get_recommendations(restaurant_name)
    return render_template('index.html', restaurant_name=restaurant_name, recommendations=recommendations, restaurant_data=restaurant_data)

if __name__ == '__main__':
    app.run(debug=True)
```

Showcasing The UI



Restaurant Recommendation System

About

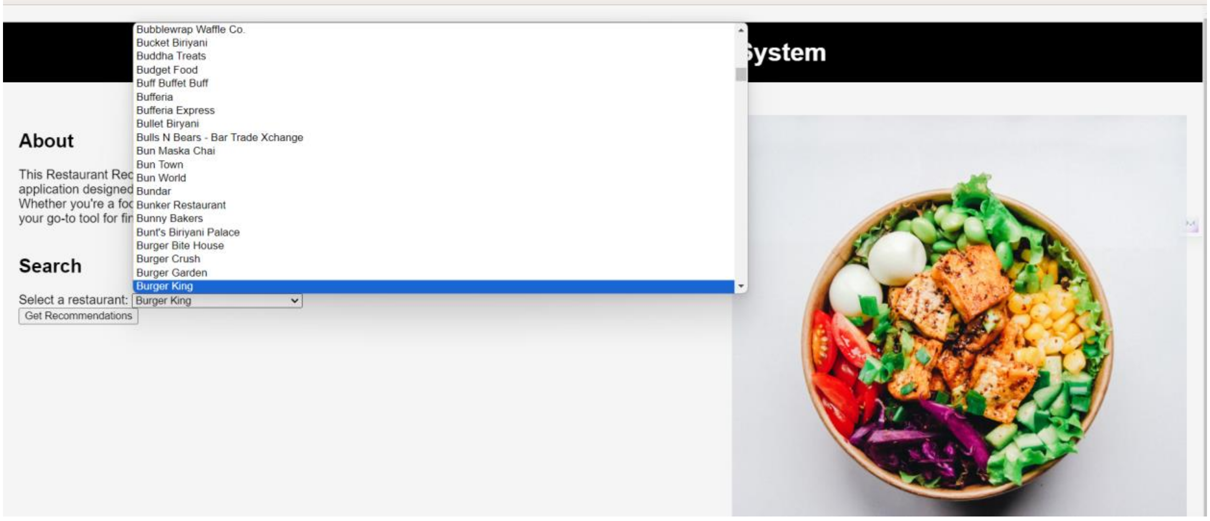
This Restaurant Recommendation System, created by Devina Goel and Prateek Choudhary, is a user-friendly web application designed to help users discover restaurants that match their preferences based on a given restaurant name. Whether you're a food enthusiast looking for a new dining experience or a traveler in an unfamiliar city, this application is your go-to tool for finding similar restaurants, complete with cost and rating information.

Search

Select a restaurant: #FeelTheROLL

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

I. Checking recommendation for the restaurant: ‘Burger King’



ystem

About

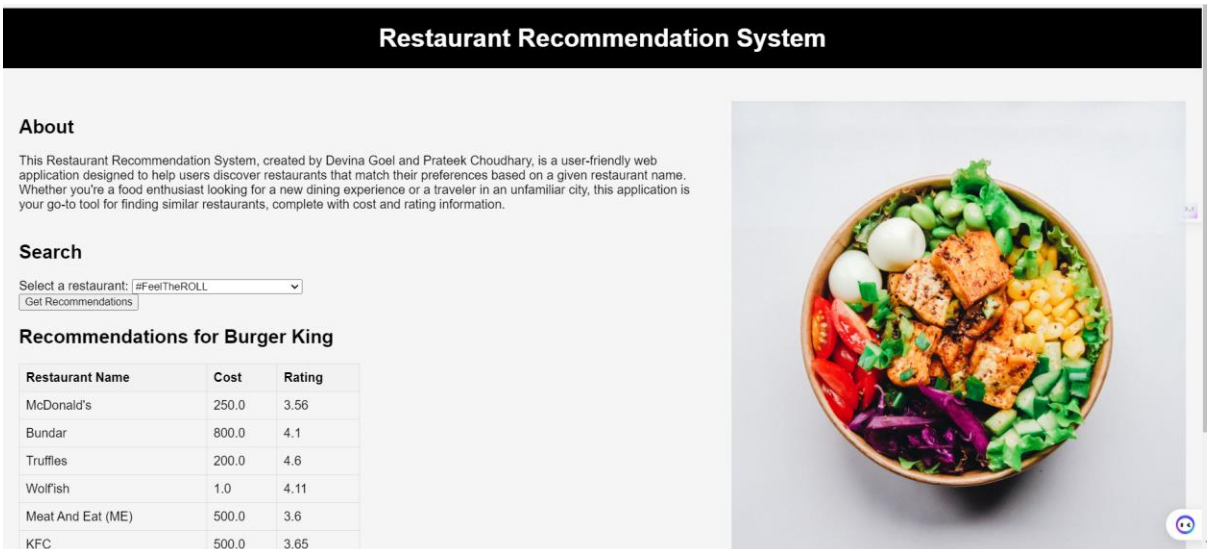
This Restaurant Rec
application designed
Whether you're a foc
your go-to tool for fir

Bubblewrap Waffle Co.
Bucket Biriyani
Buddha Treats
Budget Food
Buff Buffet Buff
Bufferia
Bufferia Express
Bullet Biryani
Bulls N Bears - Bar Trade Xchange
Bun Maska Chai
Bun Town
Bun World
Bundar
Bunker Restaurant
Bunny Bakers
Bunt's Biriyani Palace
Burger Bite House
Burger Crush
Burger Garden
Burger King

Search

Select a restaurant: Burger King

This is the prediction page where we will provide a restaurant name for which we will get the top recommended restaurants, which based on cuisines, mean rating (out of 5), cost in thousands.



Restaurant Recommendation System

About

This Restaurant Recommendation System, created by Devina Goel and Prateek Choudhary, is a user-friendly web application designed to help users discover restaurants that match their preferences based on a given restaurant name. Whether you're a food enthusiast looking for a new dining experience or a traveler in an unfamiliar city, this application is your go-to tool for finding similar restaurants, complete with cost and rating information.

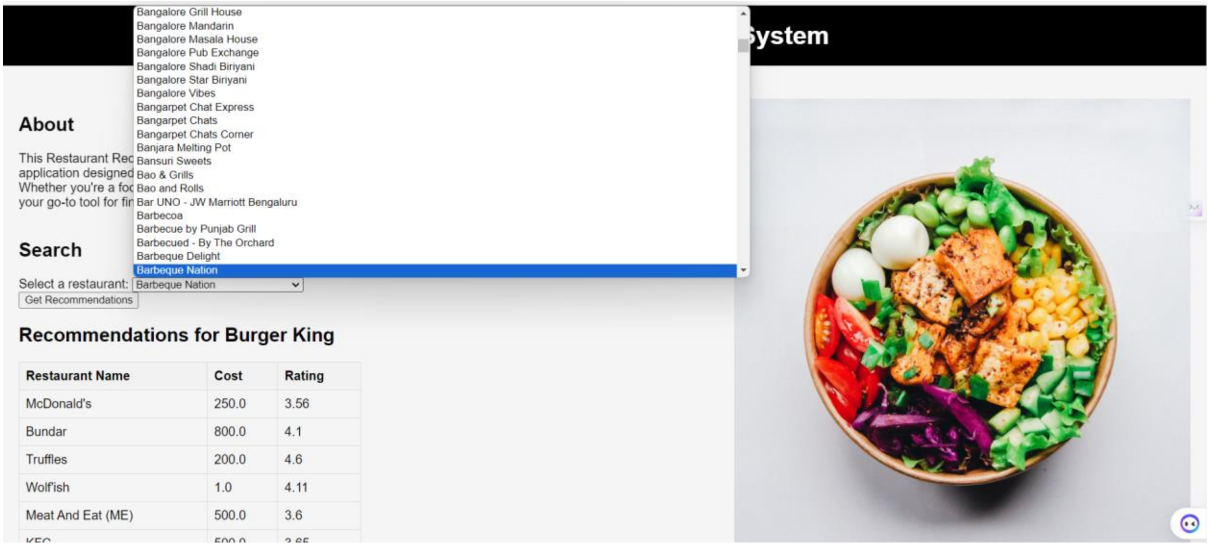
Search

Select a restaurant: #FeelTheROLL

Recommendations for Burger King

Restaurant Name	Cost	Rating
McDonald's	250.0	3.56
Bundar	800.0	4.1
Truffles	200.0	4.6
Wolf'ish	1.0	4.11
Meat And Eat (ME)	500.0	3.6
KFC	500.0	3.65

II. Checking recommendation for the restaurant ‘Barbeque Nation’



ystem

About

This Restaurant Rec
application designed
Whether you're a foc
your go-to tool for fir

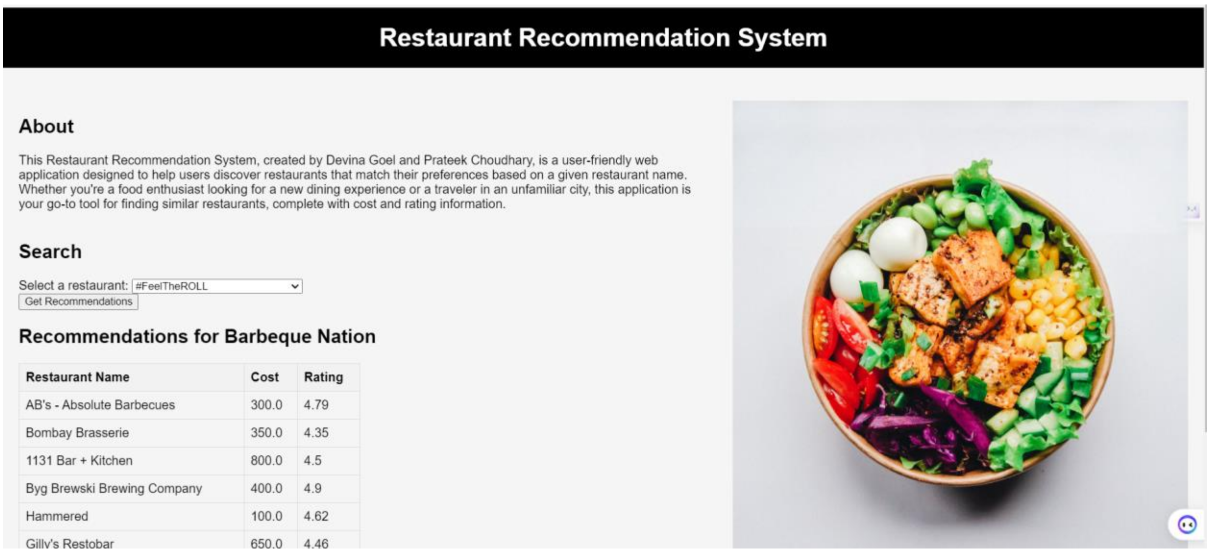
Search

Select a restaurant: Barbeque Nation

Recommendations for Burger King

Restaurant Name	Cost	Rating
McDonald's	250.0	3.56
Bundar	800.0	4.1
Truffles	200.0	4.6
Wolf'ish	1.0	4.11
Meat And Eat (ME)	500.0	3.6
KFC	500.0	3.65

And here is the recommendation



Restaurant Recommendation System

About

This Restaurant Recommendation System, created by Devina Goel and Prateek Choudhary, is a user-friendly web application designed to help users discover restaurants that match their preferences based on a given restaurant name. Whether you're a food enthusiast looking for a new dining experience or a traveler in an unfamiliar city, this application is your go-to tool for finding similar restaurants, complete with cost and rating information.

Search

Select a restaurant: #FeelTheROLL

Recommendations for Barbeque Nation

Restaurant Name	Cost	Rating
AB's - Absolute Barbecues	300.0	4.79
Bombay Brasserie	350.0	4.35
1131 Bar + Kitchen	800.0	4.5
Byg Brewski Brewing Company	400.0	4.9
Hammered	100.0	4.62
Gilly's Restobar	650.0	4.46

Finally, the prediction for the given restaurant inputs is shown.