Restaurant Recommender System for Pune City

CT-21013 DATA SCIENCE LABORATORY

Anjali Dofe | Tejal Khairnar | Yash Dhake

111903137 | 111903157 | 111903160

INTRODUCTION:

In today's digital age, we are bombarded with number of options for any given activity. These options are so vast that analysing each of them and making an informed decision becomes impossible. People casually decide to look for a similar restaurant all the time because they are addicted to a specific category of food. People who rarely use restaurants would prefer to have the most rated restaurants nearby them and all this could be easily handled by a recommendation system. This project is designed for the residents of Pune and tourists visiting Pune. The purpose of this project is to build a recommendation system to predict which restaurant customers are most likely to order from, given the customer location, budget of the customer, the rating of the restaurant and cuisine preference of the customer.

Business Requirements:

Various studies related to recommendation systems have been carried out. A lack of concentration means that restaurants are in general very diverse in terms of cuisine, hours, categories, service and even ambience. People still heavily rely on recommendations from their social circles, or just must adventure new options themselves. Given the challenge in finding individualized and desirable restaurants, it is thus critical to the platform's success to provide customers with filtered, prioritized, and personalized recommendations. Recommendation systems are an effective way to help users to get information that is useful and in accordance with user interests.

Literature Survey:

Research-paper recommender systems: a literature survey (2015) - Joeran Beel, Bela Gipp, Stefan Langer, Corinna Breitinger

This paper reviews and talks about articles and presents some descriptive statistics, as well as a discussion about the major advancements and shortcomings and an overview of the most common recommendation concepts and approaches. The paper found that more than half of the recommendation approaches applied content-based filtering (55%). Collaborative filtering was applied by only 18% of the reviewed approaches, and graph-based recommendations by 16%.

Restaurant Recommendation System Using Customer's Data Analysis(2018) - Bhagyashree Basudkar, Shruti Bagayatkar, Meghana Chopade, Sachin Darekar

This paper attempts to understand, analyse and suggest restaurants to a particular user based on user behaviour and the restaurant rankings using Zomato's API. This paper suggests that there are many context-aware restaurant recommenders but most of them only focus on location information. Some systems use only content-based filtering whereas some use only collaborative filtering. Their application will also notify the user with the nearest restaurants when the user is in motion. They have used content-based filtering as well as collaborative filtering to make the system more effective. The users' locations are tracked using gps. The recommender system adopted a user preference model by using the features of user's visited restaurants, and the location information of users and restaurants to dynamically generate the recommendations.

"Location Based Personalized Restaurant Recommendation System for Mobile Environments", International Conference on Advances in Computing, Communications, and Informatics (ICACCI), 2013 - Anant Gupta, Kuldeep Singh

In this paper, they introduced a new mobile recommendation system for restaurants that infuses personalization, ubiquity, and location-based service to take user's dining experience to the next level. They developed a recommendation system that used machine Learning algorithm to study the user's behaviour as he/she keeps checking in restaurants through any popular social network. They also eradicated problems such as cold start problem in which the system is unable to present the user any personalized recommendations. They tackled this issue by using location history from Foursquare or Facebook.

Restaurant Recommender System Using User-Based Collaborative Filtering Approach: A Case Study at Bandung Raya Region (2019) - Alif Azhar Fakhri, Z K A Baizal, Erwin Budi Setiawan

In this study, they used a user-based collaborative filtering method. If the user wants to find a restaurant recommended by another user, then the system will be searching similarity of preferences the target user and all existing users by calculating the similarity between users and the similarity of the user attributes. Then the system searches the neighbours who have biggest similarity with the target user, so that restaurants that have been given a rating by neighbours will be recommended to target users who have not rated the restaurant. Similarity to find the proximity between users is calculated using two stages: 1) Calculating the user similarity and 2) Calculating the user attribute similarity.

A Consensus location-aware group recommender system for restaurants - Jorge Castro, Óscar Cordón, Luis Martínez

This paper proposes a location-awareness group recommender system that provides recommendations according to the location context of the group and additionally such recommendations are computed to obtain a high agreement among the group members by using a consensus reaching process.

A personalized recommender system using Machine Learning based Sentiment Analysis over social data (2016) - Meghana Ashok, Swathi Rajanna, Pradnyesh Vineet Joshi, Sowmya Kamath S

In this paper, they have designed a social framework, which extracts user's reviews, comments of restaurants and points of interest such as events and locations, to personalize and rank suggestions based on user preferences. Machine Learning and Sentiment Analysis based techniques are used for further optimizing search query results. This provides the user with quicker and more relevant data, thus avoiding irrelevant data and providing much needed personalization.

PROJECT OBJECTIVES:

- Display the top-rated restaurants that are present in a particular area.
- Fetch the current location of the user and display restaurants in proximity.
- Show the least and most expensive restaurants in the area.
- Recommend restaurants based on user's cuisine preference and budget.
- Rank restaurants with respect to user's preferences.
- Search for similarity of preferences between users and their neighbours to recommend new restaurant choices based on the user's preferences.
- Recommend unrated restaurants to users to increase customer base and exposure.

Data Required:

We require the following data:

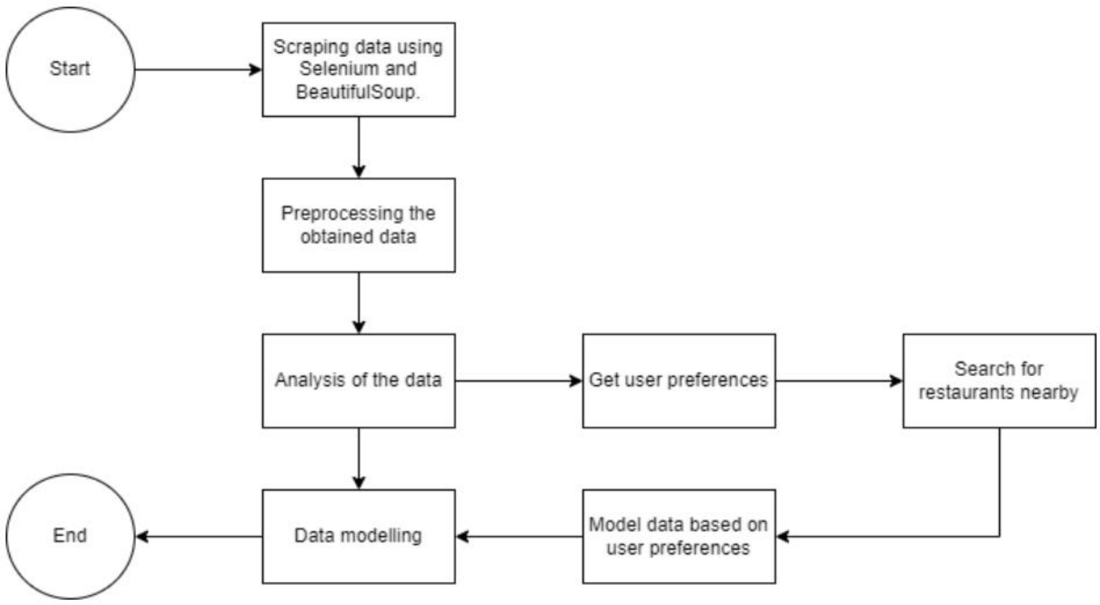
1. Location of the restaurant.

2. Cuisine offered by the restaurant.

3. Rating of the restaurant.

4. Average cost for two people.

Design Model:



Dataset Used for building the restaurant recommender system for Pune City

Readily available datasets on websites such as Kaggle were not used for this project.

The authentic dataset used for this project was obtained by performing web scraping of the website https://www.swiggy.com/city/pune by our group members. The various libraries and modules used for this purpose include Selenium and BeautifulSoup.

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from bs4 import BeautifulSoup
import pandas as pd
import requests

# binary = FirefoxBinary(r'C:\Program Files (x86)\Mozilla Firefox\firefox.exe')
# driver = webdriver.Firefox(firefox_binary=binary, executable_path = r'C:\Users\Anjali Ajay Dofe\Downloads\geckodriver-v0.28.0-win64\geckodriver.exe'
# headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.2; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71 Safari/537.36'}
# response = requests.get("https://www.swiggy.com/pune/top-rated-collection",headers=headers)
URL = 'https://www.swiggy.com/pune/top-rated-collection'
driver = webdriver.Chrome()
driver.get(URL)
# driver.get(URL)
# content = response.content
# print(content)
html=driver.page_source
# soup1 = BeautifulSoup(content,"html.parser")
soup=BeautifulSoup(html, 'lxml')
driver.close()

data1=[]
data2=[]
data3=[]
data4=[]
data5=[]

res_name=soup.find_all('div',attrs={'class':'nA6kb'})

for res_name in soup.find_all('div',attrs={'class':'nA6kb'}):
    print(res_name.text)
    data1.append(res_name.text)

for res_type in soup.find_all('div',attrs={'class':'_1gURR'}):
    print(res_type.text)
    data2.append(res_type.text)
```

EXPLORATORY DATA ANALYSIS

After gathering the data by web scraping, the next step we followed was to carry out **Exploratory Data Analysis.** It is a process of examining or understanding the data and extracting insights or main characteristics of the data. EDA is generally classified into two methods, i.e., **graphical** analysis and **non-graphical** analysis.

We examined the distribution of data and performed normalizing and scaling after handling missing values of the dataset. Data visualization was carried out using different types of charts for analysing and presenting the data. Various libraries such as NumPy, pandas and seaborn were used to perform EDA.

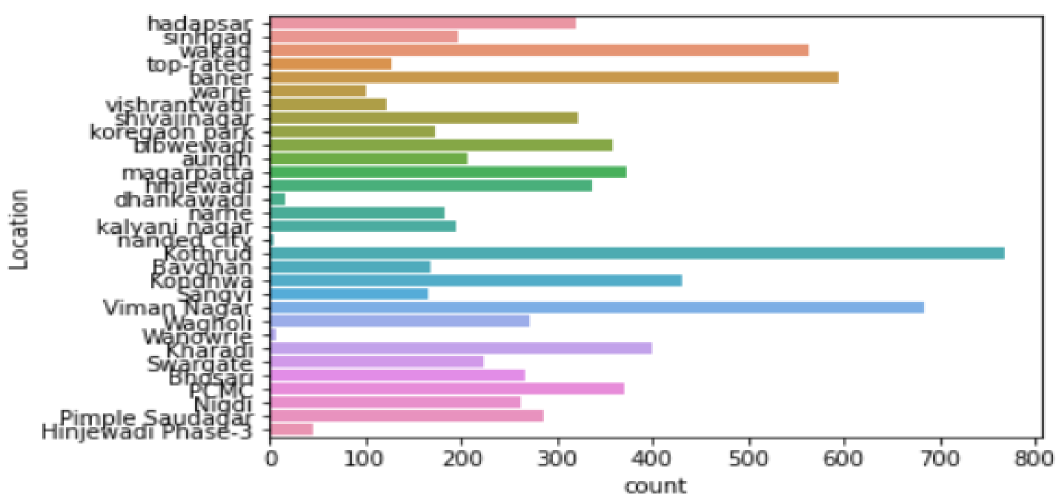
```
swiggy_data = pd.read_excel('Assgn1-Data-111903137_111903157_111903160.xlsx')
swiggy_data.head()
```

	Restaurant	Type	Rating	Delivery Time	Amount	Location
0	Haldirams Chaat	Chaat, Snacks	4.0	-- MINS	₹250 FOR TWO	hadapsar
1	Rohit food	Indian, Chinese, Desserts	0.0	- MINS•	₹1150 FOR TWO	hadapsar
2	Sahich	Maharashtrian	4.0	-- MINS	₹200 FOR TWO	hadapsar
3	SHAMS KITCHEN	South Indian	0.0	- MINS•	₹100 FOR TWO	hadapsar
4	MORYA MISAL	Snacks, Beverages	0.0	- MINS•	₹150 FOR TWO	hadapsar

NUMBER OF RESTAURANTS LOCATION-WISE

```
sns.countplot(y='Location',data=swiggy_data)
#count plot
#counting number of restaurants area/location wise
```

```
<AxesSubplot:xlabel='count', ylabel='Location'>
```



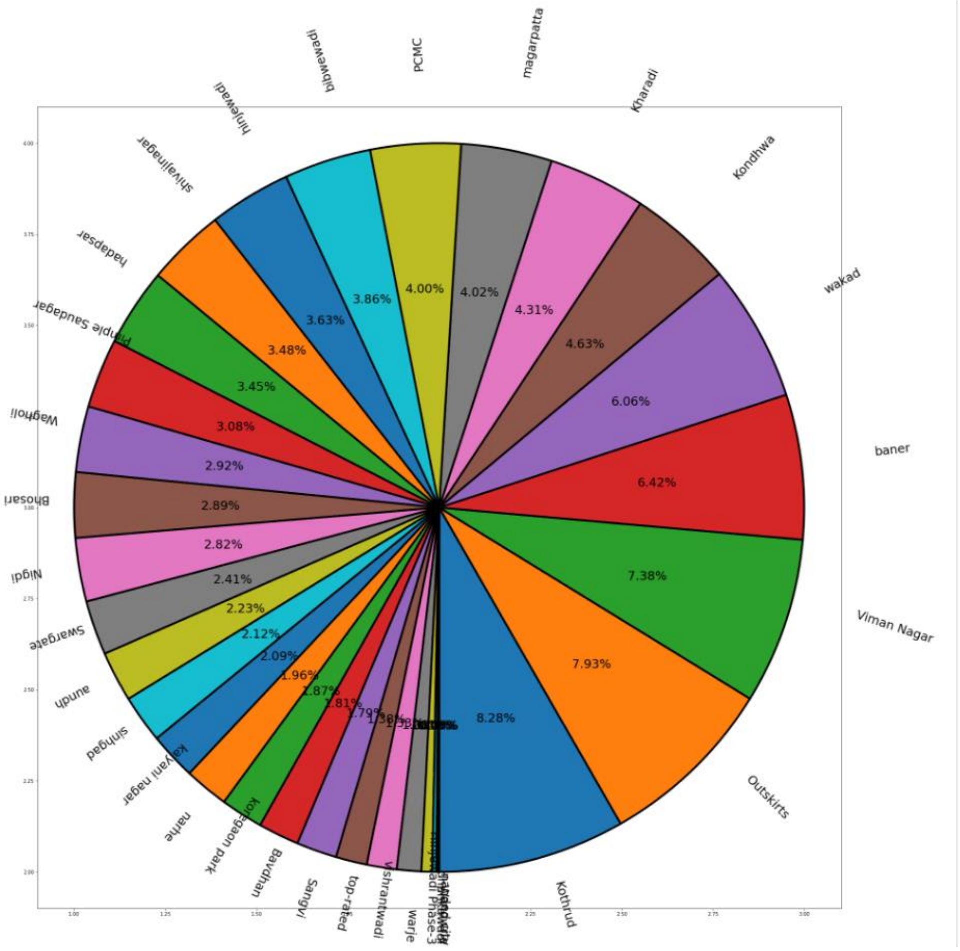
The column for DELIVERY TIME was dropped as it would be of no use in our recommender system as delivery time varies for each customer as each of them live in a different locality and is also influenced by factors such as weather conditions and traffic.

```
swiggy_data.drop('Delivery Time', axis=1,inplace=True)
# precossesing data by dropping column of delivery time as it would be different
```

```
swiggy_data.head()
```

	Restaurant	Type	Rating	Amount	Location
0	Haldirams Chaat	Chaat, Snacks	4.0	₹250 FOR TWO	hadapsar
1	Rohit food	Indian, Chinese, Desserts	0.0	₹1150 FOR TWO	hadapsar
2	Sahich	Maharashtrian	4.0	₹200 FOR TWO	hadapsar
3	SHAMS KITCHEN	South Indian	0.0	₹100 FOR TWO	hadapsar
4	MORYA MISAL	Snacks, Beverages	0.0	₹150 FOR TWO	hadapsar

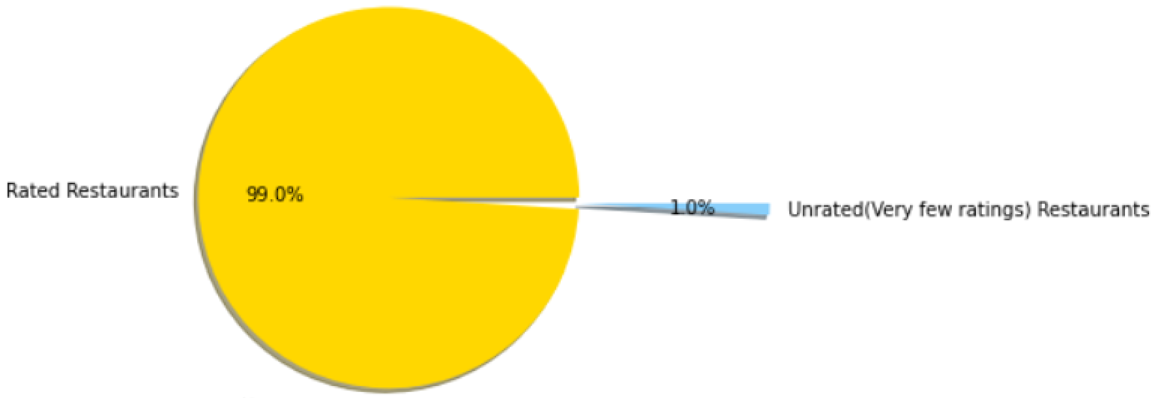
After handling all the null values again, we obtained a pictorial representation of the number of restaurants locality wise using pie chart representation.

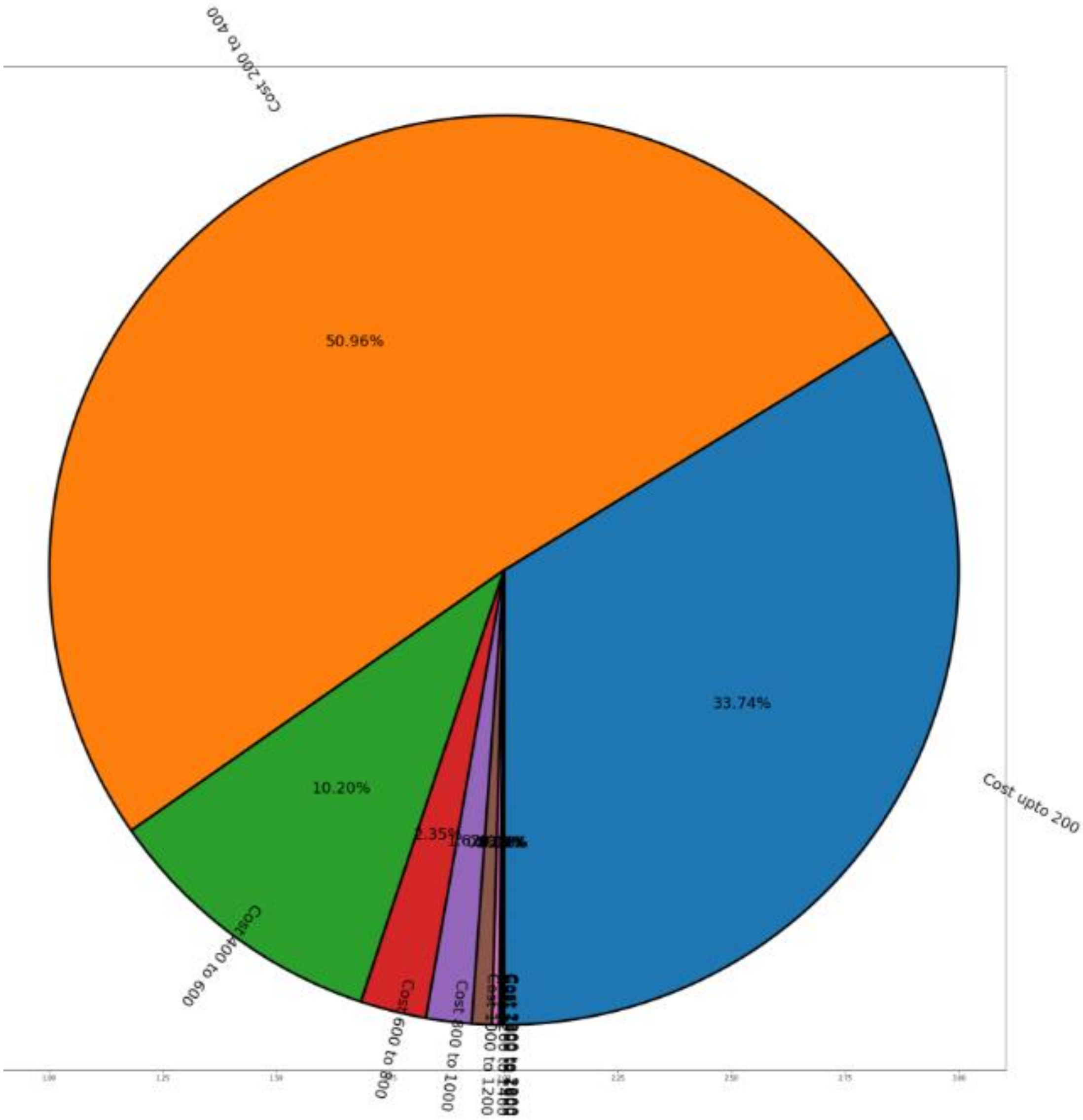


```
labels = 'Rated Restaurants','Unrated(Very few ratings) Restaurants'
sizes = [9192,91]
colors = ['gold', 'lightskyblue']
explode = (0, 1)  # explode 1st slice

# Plot
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
autopct='%1.1f%%', shadow=True)

plt.axis('equal')
plt.show()
```

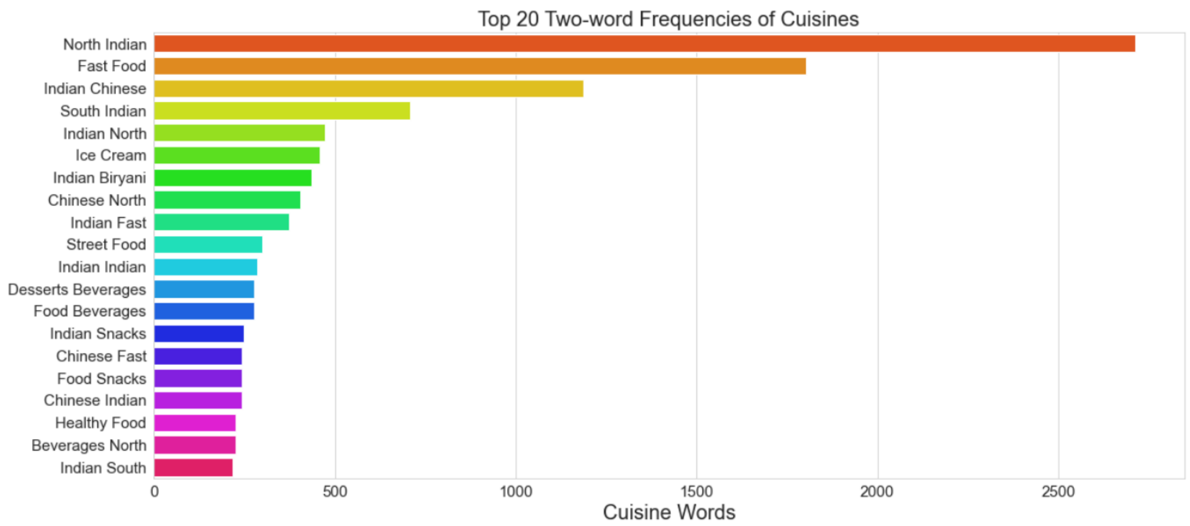




DATA MODELING and THE RECOMMENDER SYSTEM

The two most common types of recommendation systems are content-based and collaborative filtering recommender systems. In collaborative filtering, the behaviour of a group of users is used to make recommendations to other users. The recommendation is based on the preference of other users. Content-based systems are based on the idea that if you liked a certain item, you are most likely to like something that is like it.

In our project we have tried to have elements of both content-based filtering as well as collaborative filtering.




```
cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
print(cosine_similarities)
```

```
[[1.         0.         0.04743929 ... 0.03849062 0.         0.02115607]
 [0.         1.         0.         ... 0.         0.         0.        ]
 [0.04743929 0.         1.         ... 0.         0.         0.        ]
 ...
 [0.03849062 0.         0.         ... 1.         0.07766594 0.1519654 ]
 [0.         0.         0.         ... 0.07766594 1.         0.15819526]
 [0.02115607 0.         0.         ... 0.1519654  0.15819526 1.        ]]
```

The matrix Factorization we used were using Tf-Idf matrix values to carry out content-based filtering.

```python
recommend('ANNAPURNA TIFFIN')
```

TOP 10 RESTAURANTS LIKE ANNAPURNA TIFFIN WITH SIMILAR REVIEWS:

	Cost per person	Type	Rating
Relax Family Garden Resto Bar	200.0	North Indian South Indian Maharashtrian	4.2
Mumma's Kitchen(Karve Nagar)	100.0	North Indian Maharashtrian	4.1
ANNAPURNA TIFFIN	100.0	South Indian Andhra North Indian Maharashtrian...	4.0
The Biryani Experiment	250.0	Biryani Hyderabadi Indian Kebabs North Indian ...	4.0
The Taste Of Capital	150.0	North Indian Sweets Beverages	3.9
Swanandi Foods	100.0	South Indian Maharashtrian Fast Food	3.9
WORLD BIRYANI HOUSE	250.0	Biryani Sweets Beverages	3.9
Karan's Kitchen	150.0	Chinese North Indian Maharashtrian	3.9
Moonlight Kitchen	100.0	Chinese North Indian Maharashtrian	3.9
DHABA CURRY	150.0	North Indian Biryani South Indian Maharashtrian	3.9

Cheap unrated restaurants

```python
cheap_rest=swiggy_data[['Restaurant','Cost per person', 'Location','Type','Rating']]
cheap_rest=cheap_rest[(cheap_rest['Cost per person'] <1000) & ( ( cheap_rest['Rating'] == 0 ))]
cheap_rest.head(10)
```

	Restaurant	Cost per person	Location	Type	Rating
20	LeanCrust Pizza - ThinCrust Experts	175.0	hadapsar	Pizzas Italian Healthy Food	0.0
112	Hotel Red Chilly	200.0	hadapsar	Chinese North Indian	0.0
128	Chopsticks Chinese	200.0	hadapsar	North Indian Chinese	0.0
173	MADHURAM	75.0	hadapsar	South Indian	0.0
411	Exotic Chinese	75.0	sinhgad	Chinese	0.0
567	Upsouth	125.0	wakad	South Indian Fast Food	0.0
609	Patel's Chhappanbhog	200.0	wakad	Indian Chinese North Indian Punjabi Biryani Ju...	0.0
671	Combo's Kitchen	100.0	wakad	North Indian Combo Chinese	0.0
698	Jumboking-Indian Burger	75.0	wakad	Street Food Fast Food Beverages	0.0
1472	Hyderabadi Biryani Express	100.0	baner	Chinese Biryani	0.0

We also had one of our objectives to recommend unrated restaurants to users in order to increase their base and popularity, so people try them out too.

Extremely cheap low rated restaurants

```
cheap_rest=swiggy_data[['Restaurant','Cost per person', 'Location','Type','Rating']]
cheap_rest=cheap_rest[(cheap_rest['Cost per person'] <300) & ( ( cheap_rest['Rating'] < 3.0 ) & (cheap_rest['Rating'] > 0.0 ))]
cheap_rest.head(10)
```

	Restaurant	Cost per person	Location	Type	Rating
14	Royal Punjab	150.0	hadapsar	North Indian Chinese Biryani	2.7
104	Purepur Kolhapur Family Restaurant	200.0	hadapsar	North Indian Chinese	2.0
137	Samruddhi Fast Food	100.0	hadapsar	North Indian Chinese Beverages	2.9
451	Al Madina Biryani House	125.0	sinhgad	Biryani Kebabs Desserts	2.8
1949	Shree Sai Veg Non-Veg Restaurant	125.0	shivajinagar	North Indian Chinese Biryani Punjabi Tandoor	2.8
3113	AADD (M) AADD (B)	75.0	hinjewadi	Desserts Ice Cream	2.3
3308	Regal cake and bakers	100.0	hinjewadi	Desserts Bakery	2.9
3863	The Burger Project	175.0	Kothrud	American Beverages Desserts Snacks	2.9
3951	SK's Biryani House	100.0	Kothrud	Biryani Kebabs Chinese Tandoor Indian Maharash...	2.5
3972	Kadak Misal	100.0	Kothrud	Snacks	2.9

Extremely cheap low rated restaurants are the best for someone tight on their budget and not quite concerned with or particular about the food they want to eat.

High rated expensive restaurants for the ones who are more concerned about maintaining their social image and quality of the food.

832	Atmosphere 6	800.0	top-rated	North Indian Chinese Italian	4.4
836	Cafe Delhi Heights	1000.0	top-rated	Continental Asian Australian	4.4
1130	House of Mandarin	1050.0	Outskirts	Chinese Oriental Asian Sushi PanAsian Japanese...	4.3
2303	CONRAD PUNE	1000.0	koregaon park	Continental Italian	4.4
4511	Origins	800.0	Kothrud	Thai Indonesian Japanese Burmese	4.6
5651	Atmosphere 6	800.0	Viman Nagar	North Indian Chinese Italian	4.4
5884	Cafe Delhi Heights	1000.0	Viman Nagar	Continental Asian Australian	4.4

Also, collaborative filtering model which takes locality, cost and cuisine preferences from the user and accordingly suggests the best restaurants in that category to the customer.

```
inputloc = input("Enter your preferred locality : ")
inputprice = int(input("Enter your budget cost per person : "))
inputrating = float(input("Enter minimum rating : "))
```

```
Enter your preferred locality : Kothrud
Enter your budget cost per person : 300
Enter minimum rating : 4.0
```

```
recc_rest = swiggy_data[['Restaurant','Cost per person', 'Location','Type','Rating']]
recc_rest = cheap_rest[(recc_rest['Location'] == inputloc) & (recc_rest['Cost per person'] < inputprice) & ( ( recc_rest['Rating'] > inputratir
recc_rest.head(10)
```

	Restaurant	Cost per person	Location	Type	Rating
3854	Artinci - Indulge Guilt Free!	125.0	Kothrud	Healthy Food Keto Desserts Ice Cream Snacks Be...	4.4
3856	Katakirrr Misal	100.0	Kothrud	Maharashtrian Snacks	4.1
3859	Khandesh Express	175.0	Kothrud	North Indian	4.1
3862	McDonald's	200.0	Kothrud	American	4.3
3865	Natural Ice Cream	75.0	Kothrud	Ice Cream	4.7
3872	Shaukeen - The Complete Paan Shop	75.0	Kothrud	Desserts	4.6
3874	BIGGIES BURGER	125.0	Kothrud	Beverages French Indian American Continental C...	4.2
3880	Kalinga Veg Gourmet Kitchen	200.0	Kothrud	North Indian European Continental Fast Food Ma...	4.2
3884	Khandeshi Zatka	150.0	Kothrud	Maharashtrian North Indian	4.1
3889	Burger It Up	200.0	Kothrud	American Continental Beverages Desserts	4.3