Zomato Restaurant Recommendation System

A Project Report

Submitted by:

Sneh Singh (191B262)

Vanama Yaswanth (191B278)

Vidhi Mathur(191B282)

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHONOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY

GUNA , MADHYA PRADESH (INDIA) – 473226

Aug 2021- Jan 2022

DECLARATION

We hereby declare that the work reported in 5th semester Minor project entitled "**Zomato Restaurant Recommendation System**", in partial fulfillment for the award of the degree of B.Tech. submitted at Jaypee University of Engineering and Technology, Guna, as per the best of my knowledge and belief there is no infringement of intellectual property rights and copyright. In case of any violation, we will solely be responsible.

Signature of the Student

Sneh Singh (191B262)

Vanama Yaswanth (191B278)

Vidhi Mathur(191B282)

Department of Computer Science and Engineering,

Jaypee University of Engineering
and Technology,Raghogarh, Guna –
473226
Date: 19 Jan

CERTIFICATE

This is certify that the project titled "**Zomato Restaurant Recommendation System**" is the bona fide work carried out by Sneh Singh, Vanama Yaswanth and Vidhi Mathur. We are a Student of B. Tech (CSE) at Jaypee University of Engineering and Technology Guna (MP).

During the academic year 2021-2022 in partial fulfillment of the requirements for the award of the degree

Of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Signature of the Guide

Department of Computer Science and Engineering,

Jaypee University of Engineering
and Technology,Raghogarh, Guna –
473226
Date :19 Jan 2022

ACKNOWLEDGEMENT

We take this opportunity to express our deep gratitude and most sincere thanks to our project supervisor **Dr. Nilesh Kumar Jain** and project coordinator **Dr. Rahul Pachauri** for giving most valuable suggestion, helpful guidance and encouragement in the execution of this project work.

We would like to thank our supervisor and coordinator for guiding us. Last but not the least we are grateful to all the team members of "Zomato Restaurant Recommendation System".

Sneh Singh (191B262)

Vanama Yaswanth (191B278)

Vidhi Mathur(191B282)

Department of Computer Science and Engineering,

Jaypee University of Engineering
and Technology,Raghogarh, Guna –
473226

Date :19 Jan 2022

ABSTRACT

Technology has created an exceptional platform for growth of every kind of businesses. The emerging use of technology urges the need of use of IT is all possible aspects of business. Today hotel and restaurant business is one of the most growing business and has been helping a lot in the economy of the country. Through this project, I have collected the necessary details of some of the most popular restaurants Zomato Restaurant in Bangalore.

The project analyzes the data of rating provided by the end users and use the data to recommend foods and restaurants to the users. The recommendation is based on the feedback of different people on the food items. The recommendation is done on the basis of collaborative filtering algorithm.

Keywords: User-based Collaborative Filtering, Item-based Collaborative Filtering, Recommendation

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER1: INTRODUCTION

1.1.Background

Food is not just a necessity of life. The food we eat represents our culture, tradition, and values. The norms and values of a place can be significantly related to varieties of food available there. While visiting a place one of the most important factors we consider is the varieties of foods available there. NIDIA is equally rich in terms of food culture like every other thing, thus our food culture as equally be exploited to attract tourist.

But, one of the main problems is that people are unaware of famous restaurants and places available in a specific place. This not only applies to tourists but also to the local people. Restaurants culture has fostered in INDIA in last few years. Until a decade ago there were few numbers of hotels and restaurants which were dedicated to a smaller mass of people, especially tourists. But today, there are a large number of restaurants and hotels established in Bangalore INDIA.

With the increase in a number of restaurants people often get confused about the best-suited restaurant according to their preferences. In addition to that, people face a hard time to find out the best place and food to eat, especially when they are new to that place.

“Restaurant Recommendation System based on Collaborative Filtering” is a web-based restaurant recommendation system. The primary aim of the application is to suggest users the best food to eat on the given location based on their food preferences. The application is targeting everyone who wishes to go to a restaurant to eat.

The application takes the food preference and ratings into consideration to recommend food the users. The application uses item based collaborative filtering and user-based collaborativefiltering method to recommend the food to the users. The application takes user ratings for different food items and stores it into the database. The application then recommends food items to the users on the basis of their ratings

1.2 Problem Statement

In the past, people obtained suggestions for restaurants from friends or other conventional sources or sites. 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.

Second, the information provided by the site can often be biased; thus the information provided cannot always consider being accurate.

1.3 Objective

The main objectives of the application are:

- To collect user ratings on the food items of different Zomato restaurants in Bangalore.
- To recommend restaurants and foods to users based on their user ratings using collaborative filtering algorithm

1.4 Scope and Limitations

The scope of the applications are as follows:

- The restaurants and hotels within the valley will be listed in the application.

The limitations of the application are as follows:

- Only the registered restaurants and hotels will be listed in the application.
- Only the registered users will be allowed to rate the foods.

Users will be allowed to rate a particular food on the restaurant only once

1.5 Report Organization

Figure 1- Outline of the document

Preliminary Section	• Title Page • Abstract • Table of contents, list of figures and tables
Introduction	• Problem Statement and objectives • Scope and Limitation
Requirement Analysis and Feasibility Analysis	• Technical Overview • Limitation of Collaborative Filtering • Related Works • Requirement and Feasibility Analysis
System Design	• Data collection and preprocessing • Algorithm used and validation of Model • System Design
Implementation	• Tools used • Description of main modules • Testing
Conclusion and Recommendatio	• Conclusion and Recommendation

CHAPTER 2: REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS

2.1 Literature Review

Andrew Keen, author of The Cult of the Amateur wrote in his book, "How the Democratization of the Digital World is Assaulting Our Economy, Our Culture, and Our Values" that, the history of the web so far says that we are highly motivated to come up with ways to make senseof a world richer and more interesting than the constrained resources of the traditional media let on. True indeed, with the rapid growth and development of the Internet, sharing of knowledge, information and opinions became more comfortable. This increase has played a vital role in the development of social networking sites like Facebook, Twitter, and YouTube,etc. The growth of the internet, especially after web 2.0 has brought a lot of exposure for the business, armature artists, writers, etc. Now, authors can share their works with thousands of readers around the world. Amateur-musicians can get famous faster than ever before just to uploading their tracks. The business community has found more customers and profit from theinternet. The variety of online shops, auctions or flea markets opened up on the web (Asanov,2011).

Nevertheless, the popularity of WWW has introduced a new problem i.e. the amount of information and items got extremely huge, leading to information overload. The Web is a vast collection of completely uncontrolled heterogeneous documents. There are tremendous amounts of information on the internet which often becomes overwhelming for the user, and can be difficult for them to find the exact information they are searching for (Larry Page, 1994).

Recommender systems are tools used for filtering and sorting items and information. They are efficient tools that overcome the information overload, by providing users with the most relevant information by their interest. These systems are usually based on the user preference and rating. The ratings can either be acquired explicitly by filling up form, providing ratings or implicitly. Since the goal of a recommender system is to generate a meaningful recommendation to a group of users, the blueprint of the system depends on upon the domain and particular characteristics of data available. Additionally, the system may have access to user-specific and item-specific profile attributes such as demographics and product descriptions respectively.

Recommender systems differ in the way they analyze these data sources to develop notions of affinity between users and items which can be used to identify well-matched pairs (Melville, 2010).

There are various approaches used in recommender systems. The most common procedures used for recommender system are content based filtering and collaborative filtering.

2.1.1 Content-based filtering

Content-based filtering refers to such methods that provide recommendations by comparing representations of content describing an item to representations of content that interest the user pairs (Melville, 2010).

Music Recommendation systems in use web content-based filtering. The increase in multimedia data creates difficulty in searching information within user's desired time frame and according to the interest of the user.

Although the data processing time can be decreased by displaying results of songs which has been searched the most in past and present, this does not ensure that the results displayed matches the preference of the user.

Thus, in this case, content-based filtering can be used. It calculates the similarity between the content of an item (song) and user information, to display the result as per the preference of the user (Jong- Hun Kim, 2006).

2.1.2 Collaborative filtering

Collaborative filtering is the type of recommendation algorithm that bases its predictions and recommendations on the rating or behavior of other users in the system. The fundamental idea of collaborative filtering is to find other users in the community that share opinions.

There are two popular approaches of collaborative filtering:

A. User-based approach

Food Recommendation System uses the user ratings of other users with similar preferences to recommend a food item to a certain user. User-based recommendation algorithms firstly identify the k most similar users to the active user using the Pearson correlation or vector-space model in which each user is treated as a vector in the m-dimensional item space, and the similarities between the active user and other users are computed between the vectors. After the k most similar users have been discovered, their corresponding rows in the user-item matrix R are aggregated to identify a set of food items, C, ate by the group together with their frequency. With the set C, user-based CF techniques then recommend the top-N most frequent elements in C that the active user has not ate (Xiaoyuan Su, 2009).

B. Item-based approach

Though user- based approach is useful, it suffers from the scalability problem as the user base grows. Searching from the neighbors of a user becomes time-consuming. To extend collaborative filtering to the large user base, a more scalable version of collaborative filtering, the i.e. item based approach was introduced. In item based approach, instead of using similarities between users' rating to predict preferences, similarities between the evaluation patterns of a particular item is considered. Thus, the overall structure of this approach seems to be similar to that of content based approach to recommendation and personalization, but item similarity is deduced from user preference patterns rather than extracted from the item data. Even in its raw form, item–item CF does not fix anything: it is still necessary to find the

most similar to generate predictions and recommendations. In a system that has more users than items, it allows the neighborhood finding to be amongst the smaller of the two dimensions. The significant performance gain occurs as it lends itself well to pre-computing the similarity matrix. As, a user rates and re-rates items, their rating vector will change along with their similarity to other users. Finding similar users in advance is, therefore, complicated: a user's neighborhood is determined not only by their ratings but also by the ratings of other users, so their neighborhood can change as a result of new ratings supplied by any user in the system. For this reason, most user- based CF systems find neighborhoods at the time when predictions or recommendations are needed (Ekstrand, 2010).

2.1.3 Limitation of collaborative filtering method

A. Sparsity

Most users do not rate most items and hence the user rating matrix is typically very less. This is a problem for Collaborative Filtering systems since it decreases the probability of finding a set of users with similar ratings. This issue often occurs when a system has a very high item-to-user ratio or the system is in the initial stages of use. This issue can be mitigated by using additional domain information or making assumptions about the data generation process that allows for high-quality imputation (Melville, 2010).

B. The Cold-start Problem

New items and new users pose a significant challenge to recommender systems. Collectively these problems are referred to as the cold-start problem. The first of these problems arises in CF systems, where an item cannot be recommended unless some user has rated it before. This

issue applies not only to new items but also to obscure items, which is particularly detrimental to users with heterogeneous tastes. Since content-based approaches do not rely on ratings from other users, they can be used to produce recommendations for all items, provided attributes of the items are available. In fact, the content-based predictions of similar users can also be used to improve predictions further for the active user. The new-user problem is hard to tackle since without previous record of preferences of a user it is not possible to find similar users or to

build a content-based profile. As such, research in this area has primarily focused on effectively selecting items to be rated by a user so as to improve recommendation performance rapidly with the least user feedback. In this setting, classical techniques from active learning can be leveraged to address the task of item selection (Melville, 2010).

C. Fraud

As Recommender Systems are increasingly adopted by commercial websites, they have started to play a significant role in affecting the profitability of sellers. This has led to many unscrupulous vendors engaging in different forms of fraud to game recommender systems for their benefit. Typically, they attempt to inflate the perceived desirability of their products or lower the ratings of their competitors. These types of attack have been broadly studied as shilling attacks or profile injection attacks. Such attacks usually involve setting up dummy profiles and assume different amounts of knowledge about the system. For instance, the average attack assumes knowledge of the mean rating for each item; and the attacker assigns values randomly distributed around this average, along with a high score for the item being pushed. (Melville, 2010).

2.2 Related Works

Recommender system has been widely used in recent days, especially in the field of e-commerce. Listed below are some of the popular application based which uses recommendation algorithm.

2.2.1 Amazon.com

Amazon.com is the largest internet- based retailer of US. It uses recommendations as a targeted marketing tool in many email campaigns and on most of its websites' pages. Clicking on "Your Recommendations" link clients are directed to a page where they can filter their recommendations by product line and subject area, rate recommended products and rate their previous purchase. Our shopping cart recommendations offer product suggestions to the clients based on the items in their shopping cart (Linden, 2003).

2.2.2 Netflix

Netflix is the world's leading internet television network with over 81 million members in 190 countries. It has a massive database of TV shows, movies, documentaries, etc. It recommends videos to the user by the shows they watch, their ratings on previously watched shows, etc (Melville, 2010).

2.2.3TripAdvisor.com

TripAdvisor is one of the world's largest travel site, which enables travelers to plan and book their trip to almost every part of the world. It recommends places to the users by whether, their past travel patterns, type of trip, etc (Melville, 2010).

2.1.4 Moviefinder.com

Moviefinder.com allows customers to locate movies with a similar "mood, theme, genre or cast" to a given film. From the information page of the film in question, customers click on the Match Maker icon and are provided with the list of recommended movies, as well as links to other films by the original film's director and the main actors (Melville, 2010)

2.1.5We Predict

We Predict uses item-based collaborative filtering method. It recommends movies to customers based on their previously indicated interests. Customers enter a rating on a 5-point scale -- from A to F – for movies they have viewed (Melville, 2010).

2.3 Requirement Analysis

The functional and non-functional requirements addressed by the application are listed below in the Table 1.

Table 1-Functional and non-functional requirement

S.N	Functional Requirement	Non-functional Requirement
1.	User Registration	1. The user can register using a valid email address. 2. Only one account can be created to with an email address. 3. The username should contain at least 4 characters. 4. The username can only contain alphabets and numbers 5. The password length should be a minimum of 8 letters with at least one digit. 6. The user should specify if they are vegetarian or nonvegetarian.
2.	User login	The user can login with an email username and password.

3.	Rate food	1. The user can rate food on the scale of 1 to 5 , 1 representing the worst and 5 representing the best taste and quality. 2. The user can rate a specific food only once.

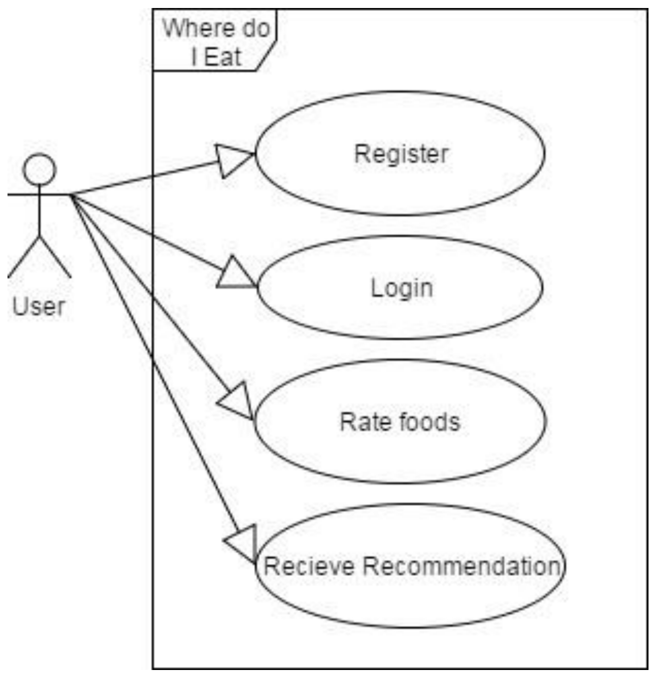


Figure 2- Use case diagram

As shown in the Figure 2, the end user is allowed to create a new account by registering to the application. The email address should be unique for each user.

The registered user can log in to the application by entering their valid username and password.

Once the user logins to the application they can rate food items of different restaurants. A user can rate a specific food of the restaurant only once.

The user can receive recommendation either on the basis of user-based CF or item-based CF based on the choice of the user. The user needs to have at least 20 ratings for getting the recommendation based on item-based CF

2.4 Feasibility Analysis

2.4.1 Technical feasibility

Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps.

2.4.2 Operational feasibility

The application used 2-tier architecture. The clients of the applications are the end users who rate the food items and receive recommendations. The server keeps the records of all users, restaurants, food items and the user ratings and responds to the client’s request.

The application can be accessed from anywhere with an internet connection. It is easy to use. Thus, it was determined to be operationally feasible.

Step 1: The activity specification table is constructed with Work Breakdown Structure.

Table 2- Work breakdown structure

Activity	**Time (weeks)**	**Predecessor**
Data Collection (A)	1	-
Database Design (B)	1	-
Data Preprocessing (C)	1	A, B
User Registration and Recording Data (D)	3	C
Implement User Based Filtering (E)	2	D
Implement Item Based Filtering (F)	2	E
Front End Design (G)	4	E
Testing (H)	5	G
Documentation (I)	9	D

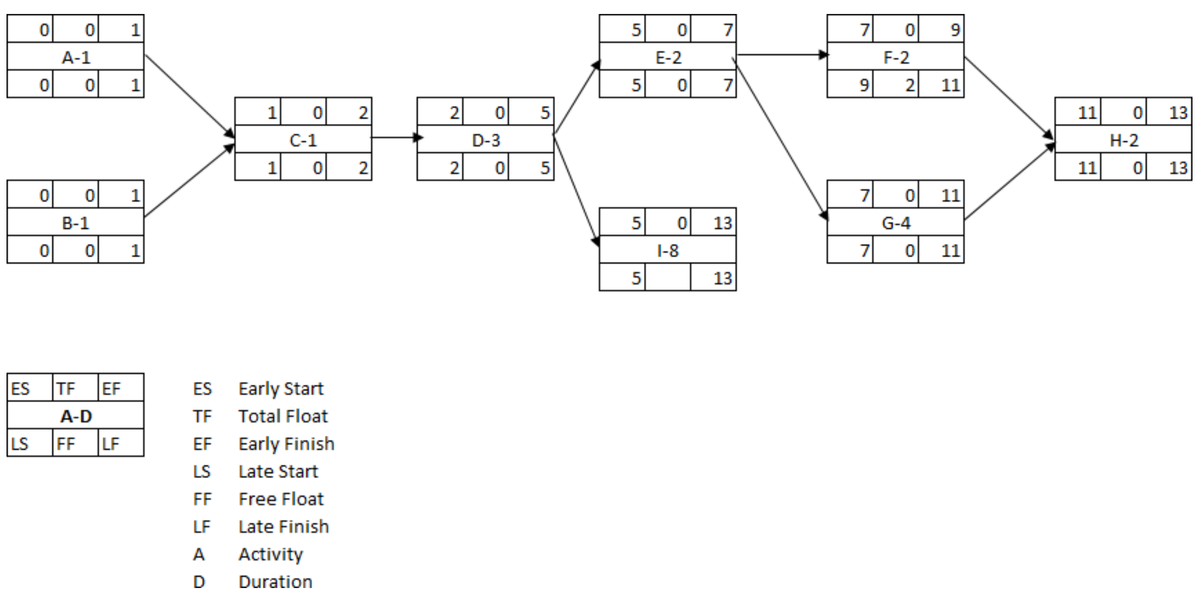


Figure 3- Activity network diagram

From the Figure 3 we can see that the application was completed in 13 weeks which is within15 weeks of a semester. Hence, the project was determined to be feasible in terms of schedule.All the activities except activity F are critical.

CHAPTER 3: SYSTEM DESIGN

3.1 Methodology

Since the requirements of the application were clear and waterfall model was used to develop the application. The detailed methodology used to develop the application are described in thefollowing subsections.

3.1.1 Data collection

The list of restaurants and their food menu were collected using web scraping from the websites, www.kaggle.com.

A survey was conducted to know the foods and restaurant commonly preferred by people.

This dataset contains reviews of restaurants from Zomato. The data contains **~51,000 records and 17 columns**. Reviews include user information, ratings, and a plain text of the review. There are approx **6500 unique restaurants** in our data.
The data set has **attributes** like- URL, address, rate, votes, location, cuisines, approx cost and reviews list But, we will be mainly making use of features like URL, Cuisine, Rate, Reviews list etc

3.1.2 Data preprocessing:

I already notice some issues with the data that I'll need to address before analyzing it — removing duplicate values, removing NaN values, changing column names, transforming data, cleaning the text columns. This is an extensive dataset that most likely requires quite a lot of cleaning

	types	counts	uniques	nulls	min
dish_liked	object	23639	5272	28078	NaN
rate	object	43942	65	7775	NaN
phone	object	50509	14927	1208	NaN
approx_cost(for two people)	object	51371	71	346	NaN
rest_type	object	51490	94	227	NaN
cuisines	object	51672	2724	45	NaN
location	object	51696	94	21	NaN
address	object	51717	11495	0	# 31, 7th Cross, Opposite Canara Bank, Domlur,...
url	object	51717	51717	0	https://www.zomato.com/24-7RestaurantBangalore...
reviews_list	object	51717	22513	0	[('Rated 1.0', "RATED\n ... Just now I tasted...
menu_item	object	51717	9098	0	["A Simple Man's Breakfast", "The Gentleman's ...
online_order	object	51717	2	0	No
name	object	51717	8792	0	#FeelTheROLL
listed_in(type)	object	51717	7	0	Buffet
listed_in(city)	object	51717	30	0	BTM
book_table	object	51717	2	0	No
votes	int64	51717	2328	0	0

Figure 4- Sample data of rating table

3.1.3 Algorithms used

Since both individual collaborative filtering have their own limitations which can be minimized in an application if both of the algorithms are used. Both the collaborativetechniques i.e. User-based Collaborative Filtering and Item-Based Filtering were used in the application.

A . User-Based Collaborative Filtering:

This filtering methodology is used torecommend items to the user based on the rating of other users having similarpreferences. The algorithm works perfectly fine when the number of users and items are less. But, when the number of items starts increasing problems of data sparsity occurs

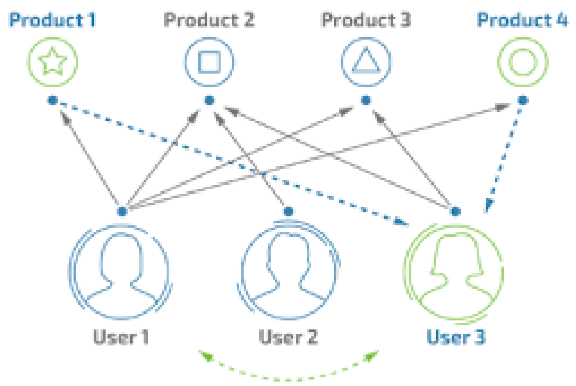


Figure 5- User-based collaborative filtering

Algorithm

1. Collect preference of all the users listing the ratings of users on different food items.
2. Calculate similarity between the users using Pearson Correlation Coefficient.

$$r = \frac{n(\sum xy) - (\sum x).(\sum y)}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}}$$

where,

n = number of pairs of scores

$\sum xy$ = sum of products of paired scores

$\sum x$ = sum of x scores

$\sum y$ = sum of y scores

$\sum x^2$ = sum of squared of x scores

$\sum y^2$ = sum of squared of y scores

Sort the similarity scores between the users such that the users in descending order.

3. Produce weighted score that ranks the users by taking the multiplying their ratings ofdifferent foods with the similarity score.

A. **Item-based Collaborative Filtering:** This filtering methodology is used to recommend items to the users based on their previous ratings. This method is similar to user-based collaborative filtering, except that the similarity between items is calculated instead of similarity between users.

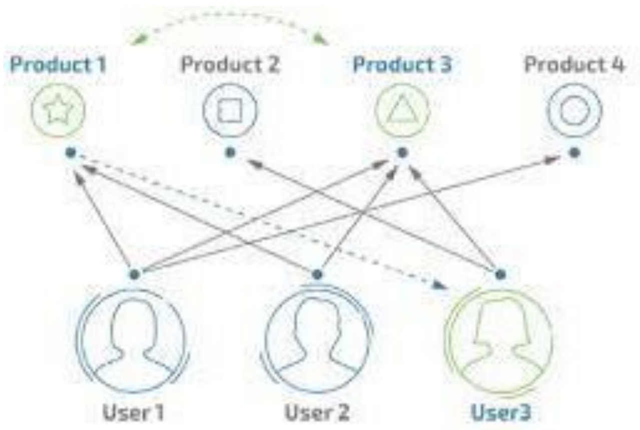


Figure 6- Item-based collaborative filtering

Algorithm

1. List all the items with the given to them by different users.
2. Calculate similarity between the items using Pearson Correlation Coefficient.

$$r = \frac{n(\sum xy) - (\sum x).(\sum y)}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}}$$

where,

n = number of pairs of scores

$\sum xy$ = sum of products of paired scores

$\sum x$ = sum of x scores

$\sum y$ = sum of y scores

$\sum x^2$ = sum of squared of x scores

$\sum y^2$ = sum of squared of y scores

3. Sort the similarity between the items such that the items in descending order.
4. Produce weighted scores that rank the items by multiplying the ratings by differentusers similarity score.

3.1.4 Validation of model

The validation of the model was done using Precision and Recall model.

$$\text{Precision}= \frac{tp}{tp+fn}$$

$$\text{Recall} = \frac{tp}{tp+fp}$$

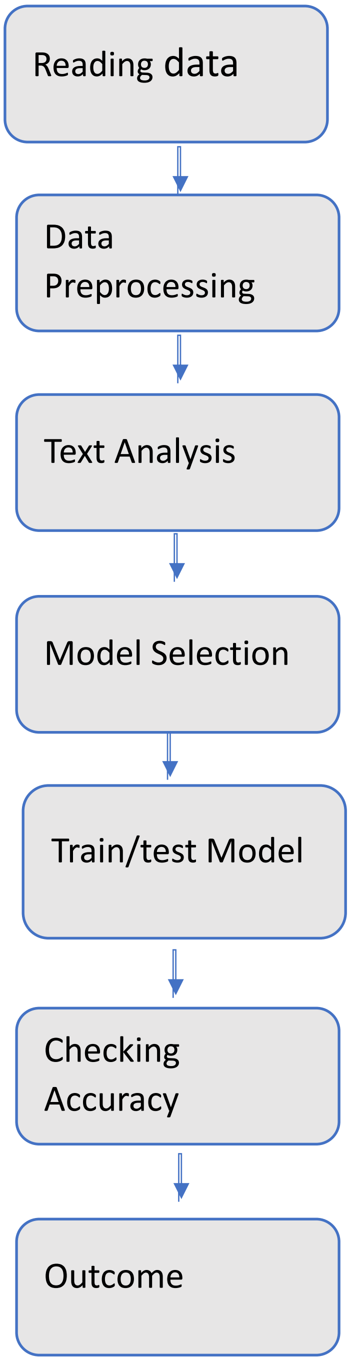
Where,

tp = true positive fn = false

positivefp = false positive

3.2 System Design

3.2.1 Flowchart


Reading data
Data Preprocessing
Text Analysis
Model Selection
Train/test Model
Checking Accuracy
Outcome

3.2.2 State diagram

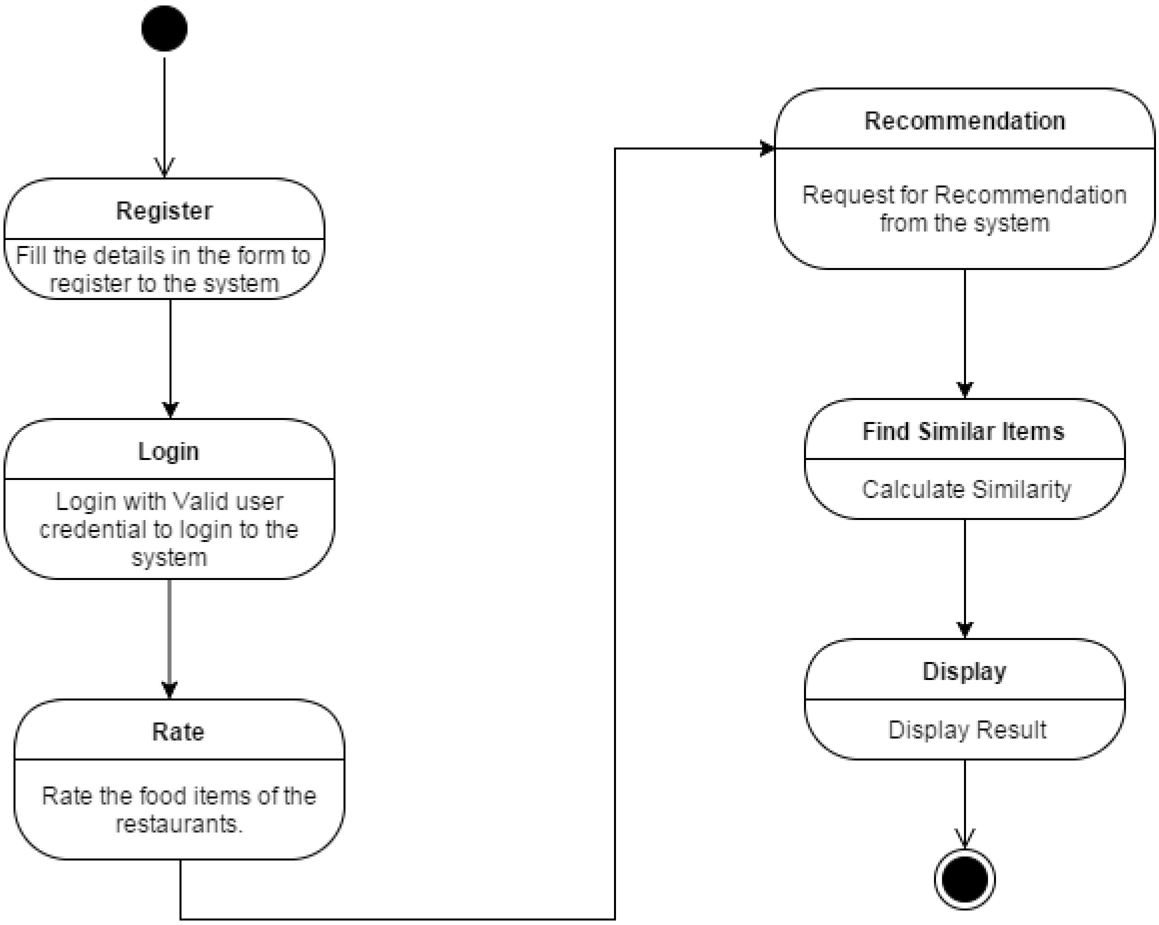


Figure 8- State diagram

The Figure 8 explains states of the application. At first, when the user opens the application for the first time, they need to register to the application. Then they are required to log into the application with their valid user credentials. The user can now rate different food items ofdifferent restaurants. The ratings of the users are saved into the database.

The user can also choose to get recommendations. When the user searches for therecommendation, the similarity is calculated and the results are displayed on the screen.

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1 Tools used

The application is based on STREAMLIT framework. It uses python programming language in back-end and Streamlit in front-end. The algorithms were implemented in python.
MS Excel was used for data preprocessing and draw.io was used as case tool

4.1.2 Description of main modules

One of the major classes used in the implementation of the application is the class that calculates the similarity between the users/ items using Pearson Correlation Coefficient.

In user-based collaborative filtering, once the users have been listed with the items that they have rated, Pearson correlation coefficient is used to calculate the similarity betweenthe users.
Similarly, in item-based collaborative filtering, Pearson Correlation Coefficient is used to calculate the similarity between the items based on the users who have rated the items.

```
def sim_pearson(prefs,p1,p2):
# Get the list of mutually rated itemssi={}
for item in prefs[p1]:
        if item in prefs[p2]: si[item]=1# Find
the number of elements n=len(si)
# if they are no ratings in common, return 0if n==0:
return 0
# Add up all the preferences
sum1=sum([prefs[p1][it] for it in si])
sum2=sum([prefs[p2][it] for it in si])# Sum
up the squares
sum1Sq=sum([pow(prefs[p1][it],2)    for    it    in    si])
sum2Sq=sum([pow(prefs[p2][it],2) for it in si])# Sum up
the products
pSum=sum([prefs[p1][it]*prefs[p2][it] for it in si])#
Calculate Pearson score
num=pSum-(sum1*sum2/n)
den=sqrt((sum1Sq-pow(sum1,2)/n)*(sum2Sq-pow(sum2,2)/n))if
den==0: return 0
r=num/denreturn r
```

4.2 Testing

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

	cuisines	Mean Rating	cost
Gokul Kuteera	North Indian, Chinese, South Indian	3.71	650.0
Cinnamon	North Indian, Chinese, Biryani	3.62	550.0
Altaf'S Chillies Restaurant	North Indian, Chinese	3.61	500.0
Pallavi Restaurant	Biryani, Chinese, Andhra	3.58	500.0
Prasiddhi Food Corner	Fast Food, North Indian, South Indian	3.45	200.0
Swad Restaurant	Chinese, North Indian	3.45	550.0
Hyderabad Cuisine Multicuisine Restaurant	North Indian, Chinese, Biryani	3.45	700.0
Chef In	Biryani, North Indian, Chinese	3.32	500.0
Swad 'E' Punjab	North Indian, Chinese, Mughlai	3.32	500.0
Punjabi Dawat	North Indian, Chinese	2.42	400.0

```
recommend('Pai Vihar')
TOP 10 RESTAURANTS LIKE Pai Vihar WITH SIMILAR REVIEWS:
```

	cuisines	Mean Rating	cost
Atithi	North Indian	3.63	750.0
Cinnamon	North Indian, Asian, Continental	3.62	1.0
Samosa Singh	Street Food, Beverages	3.60	150.0
New Friends	North Indian, Continental, Chinese, Steak	3.58	900.0

	cuisines	Mean Rating	cost
Prasiddhi Food Corner	Fast Food, North Indian, South Indian	3.45	200.0
Shrusti Coffee	Cafe, South Indian	3.45	150.0
Shanthi Sagar	South Indian, Chinese, North Indian	3.44	
Shanthi Sagar	South Indian	3.44	250.0
Mayura Sagar	Chinese, North Indian, South Indian	3.32	250.0
The Diner	North Indian, Chinese, Andhra, Biryani	3.32	700.0

CHAPTER 5: CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Content-based is a great model to start with for recommending restaurants for a user but it lacks variations in recommended restaurants. Most of the restaurants that are recommended will be of the same nature. It is of better use when we want to find similar restaurants as users can have a very diverse taste and can like a different type of restaurants having completely different cuisines. But the good part is you don't need other users data and so no cold start problem i.e. not having sufficient data to recommend items. This is normally faced by a new application that is still gathering data

5.2 Recommendations

The data used on the application is solely based on the data extracted web scraping. In order to commercially use the product, it is important to collect data of all the zomato restaurants in the Bangalore. Furthermore, since the use of mobile phones is huge, the application will be more effective if built in mobile platform.

CHAPTER 6: REFERENCES

1 https://www.kaggle.com/learn/intermediate-machine-learning

2 https://towardsdatascience.com/

3 https://numpy.org/doc/stable/

4 https://pandas.pydata.org/docs/

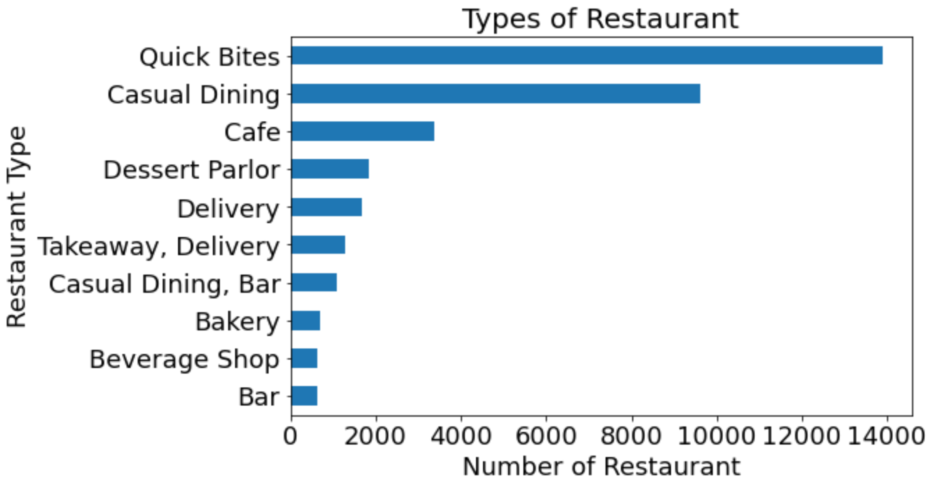
5 https://toloka.ai/ml/natural-language-processing?utm_source=google&utm_medium=cpc&utm_campaign=Search_Worldwide_eng_Desktop_B2B_Requesters_toloka|15008163823&utm_term=natural%20language%20processing&utm_content=k50id|kwd-12518176|cid|15008163823|aid|554564199788|gid|130337878922|pos||src|g_|dvc|c|reg|9298810|rin||&k50id=130337878922|kwd-12518176&gclid=CjwKCAiA55mPBhBOEiwANmzoQjnIY7CAR3ASRviGn0-G0jBPSzBOCTNsZDiU-Q_Lz_6GDUhUsidyDRoC5XwQAvD_BwE

6 https://docs.streamlit.io/

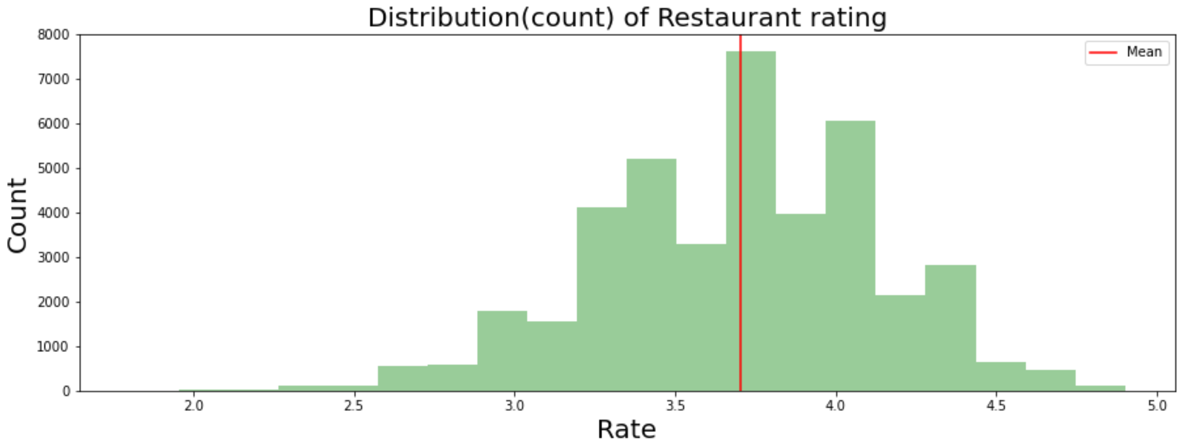
CHAPTER 7: APPENDICES

Sample data of user table

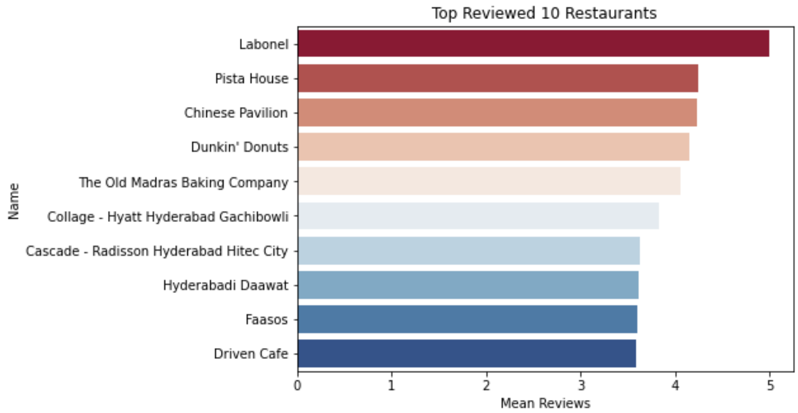
- Types of restaurant



Distribution of Restaurant Rating



- Top 10 Rated Restaurants



Top 15-word frequency for cuisines

```
lst = get_top_words(zomato_df['cuisines'], 15, (2,2))df_words =
pd.DataFrame(lst, columns=['Word', 'Count'])plt.figure(figsize=(7,6))
sns.barplot(data=df_words, x='Count', y='Word')
plt.title('Word Couple Frequency for Cuisines');
```