DWIT COLLEGE

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Institute of Science and Technology

RESTAURAURANT RECOMMENDATION SYSTEM BASED ON COLLBORATIVE FILTERING

A PROJECT REPORT

Submitted to

Department of Computer Science and Information Technology

DWIT College

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Submitted by

Prabina Neupane

August, 2016

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

SUPERVISOR'S RECOMENDATION

I hereby recommend that this project prepared under my supervision by PRABINA NEUPANE entitled **"RESTAURAURANT RECOMMENDATION SYSTEM BASED ON COLLBORATIVE FILTERING"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

…………………………………………

Asst Prof. Sarbin Sayami

IOST

Tribhuvan University

DWIT College

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

LETTER OF APPROVAL

This is to certify that this project prepared by PRABINA NEUPANE entitled **“RESTAURAURANT RECOMMENDATION SYSTEM BASED ON COLLBORATIVE FILTERING”** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

…………………………………… Sarbin Sayami [Supervisor] Assistant Professor IOST, Tribhuvan University	………………………………………… Hitesh Karki Chief Academic Officer DWIT College
………………………………………….. Jagdish Bhatta [External Examiner] IOST, Tribhuvan University	………………………………………….. Rituraj Lamsal [Internal Examiner] Lecturer DWIT College

ACKNOWLEDGMENT

This project is the result of dedication and encouragement of many individuals. My sincere and heartfelt appreciation goes to all of them. I would like to express my sincere gratitude to Deerwalk Institute of Technology for providing me with the opportunity to do this project.

I am thankful to Mr. Hitesh Karki, Chief Academic Officer, Deerwalk Institute of Technology for availing me of resources and continuous support that helped us to carry out this project. I owe my debt of gratitude to our Project Supervisor Asst. Prof Sarbin Sayami, for supervising our task and constantly motivating us with ideas regarding the project from the time of its inception. I would like to thank my teachers Mr.Bikash Balami, Mr. Narayan Chalise and Mr. Prakash Amatya for their constant support and guidance. I am also grateful to Mr. Chandan Gupta for helping me throughout the project.

Last but not the least, I am obliged to everyone who suggested and supported me in the course of my project from the beginning to the end.

Prabina Neupane

TU Exam Roll no: 1805/069

STUDENT’S DECLARATION

I hereby declare that I am the only author of this work and that no sources other than the listed have been used in this work.

... ...

Prabina Neupane

Date: August, 2016

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 in Kathmandu Valley.

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

LIST OF ABBREVIATIONS

CF: Collaborative Filtering

HTML: Hypertext Markup Language

CPM: Critical Path Method

PR: Precision and Recall

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. For example, Kirtipur and Kokhana are popular for the Newari foods, whereas Sumai Momo is renowned for the mouthwatering momo. While visiting a place one of the most important factors we consider is the varieties of foods available there. Nepal 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 Nepal 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 Nepal, Kathmandu.

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 collaborative filtering 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 restaurants in Kathmandu Valley.
- To recommend restaurants and foods to users based on their user ratings using collaborative filtering algorithm

1.4 Scope and Limitations

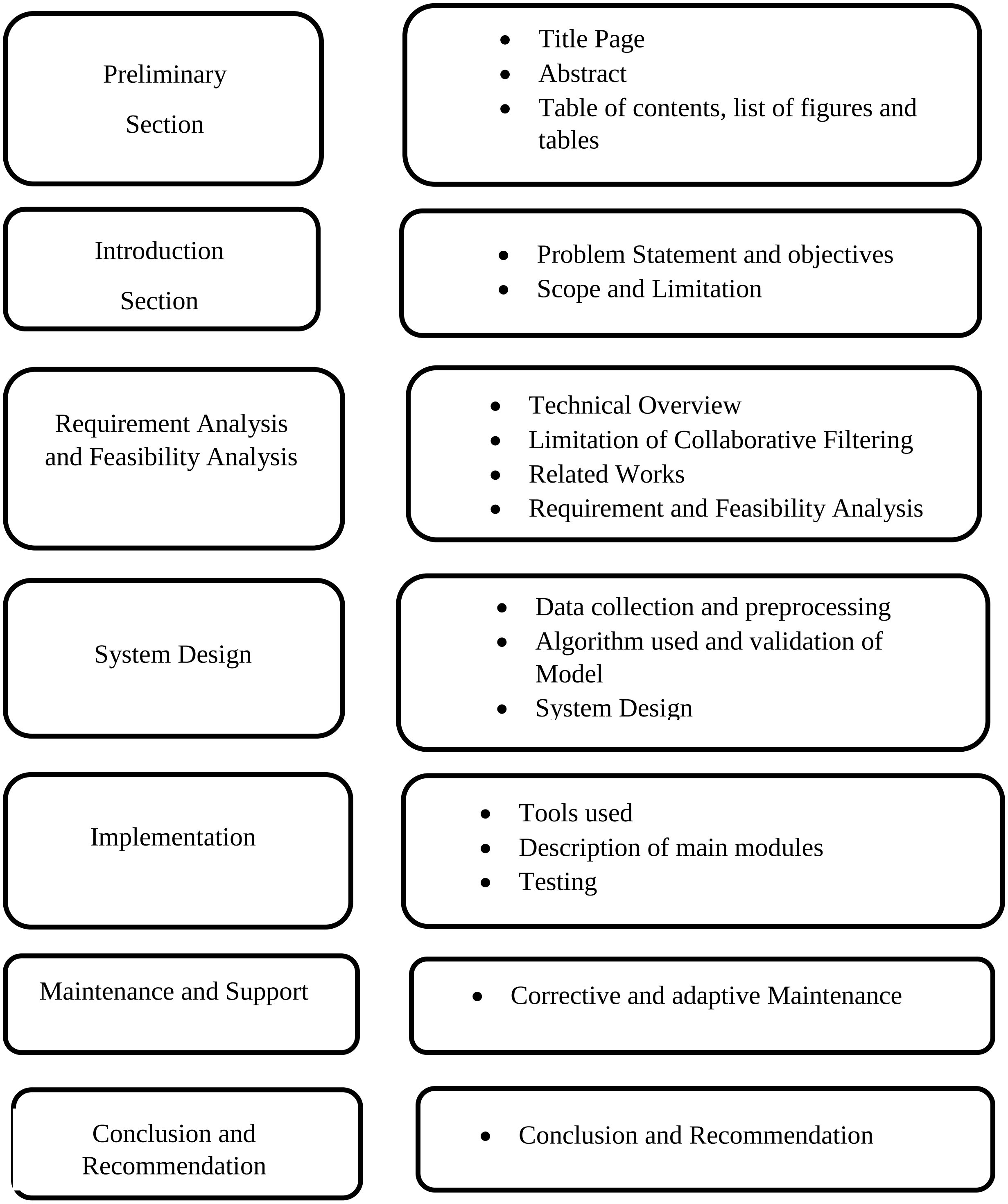
The scope of the applications are as follows:

- 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



Preliminary Section	• Title Page • Abstract • Table of contents, list of figures and tables
Introduction Section	• 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
Maintenance and Support	• Corrective and adaptive Maintenance
Conclusion and Recommendation	• Conclusion and Recommendation

Figure 1- Outline of the document

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 sense of 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 the internet. 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).

2.2.3 TripAdvisor.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).

2.2.4 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)

2.2.5 We 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).

2.2.6 YellowNepal

YellowNepal is an application available in both web and mobile platform. It provides the list of restaurants within Kathmandu valley. It only performs location based search. The user preference and ratings are not taken into consideration in the application.

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.

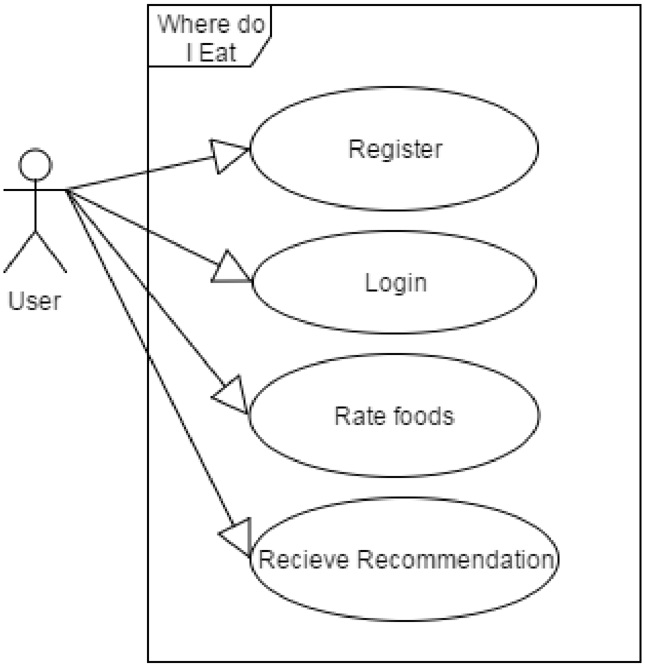


Figure 2- Use case diagram

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

The application uses HTML to display content in the browser, CSS to beautify the HTML content, and JavaScript is used for making the web page interactive. At the server side, it uses python to implement the logic, Flask web framework for dynamic web page generation and to display the predicted result in the browser as well as to handle page requests It requires a server, client, and internet connection to function properly. It supports both Windows and Linux platform for its operation. All of the technology required by the application are available and can be accessed freely, hence it was determined technically feasible.

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.

2.4.3 Schedule feasibility

The schedule feasibility analysis is carried out using the CPM method. CPM was used to identify critical tasks and calculate the interrelationship between tasks. The plan was carried out which defined critical and non-critical tasks with the goal of preventing time-frame problems and process bottlenecks. The CPM analysis was carried out as follows:

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

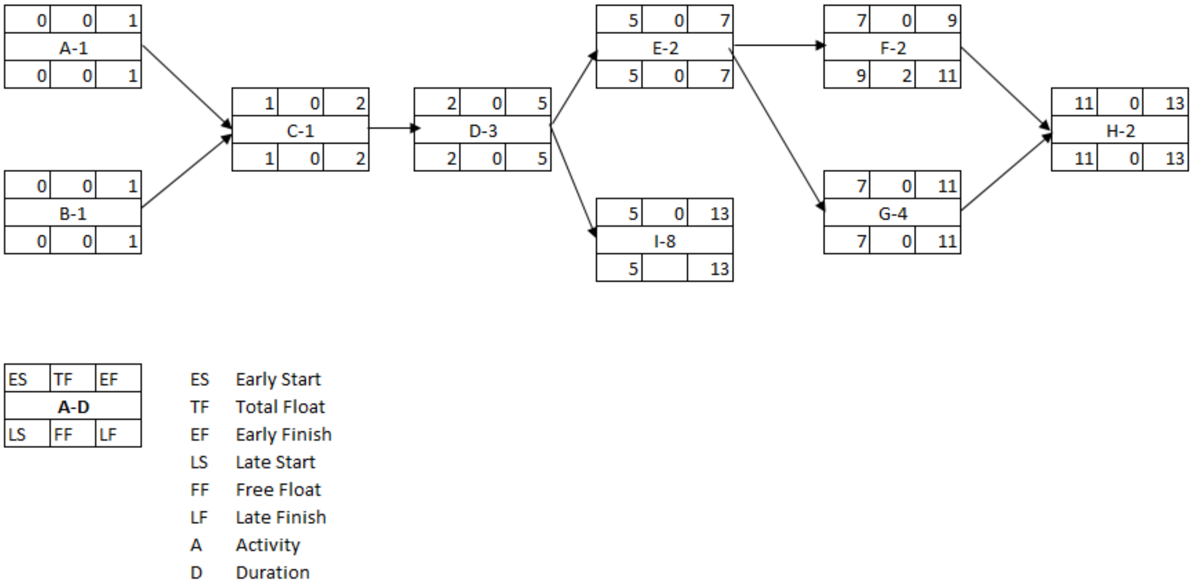


Figure 3- Activity network diagram

From the Figure 3 we can see that the application was completed in 13 weeks which is within 15 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 the following subsections.

3.1.1 Data collection

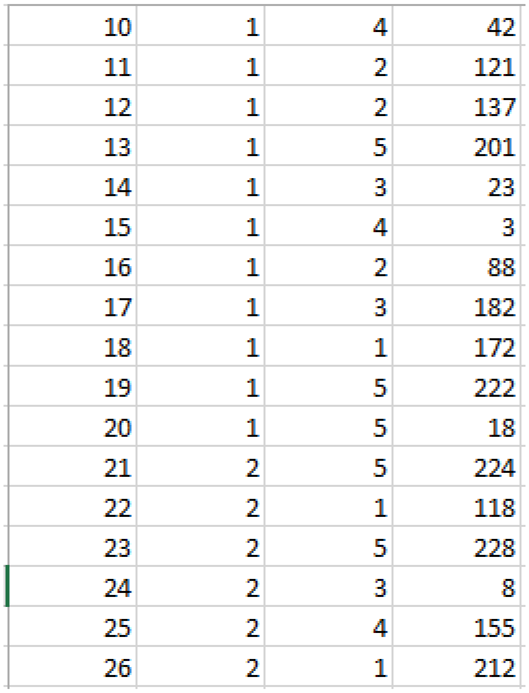
The list of restaurants and their food menu were collected using web scraping from two websites, www.foodmandu.com, and www.yellowpages.com.

A survey was conducted to know the foods and restaurant commonly preferred by people of different age group and gender. 178 responses were received in total in which 127 responses were from people of age group 15-25, 34 responses were from the age group of 25-35 and 17 were from age group 35 above.

According to the response collected, 35 unique foods and 25 restaurants were listed and recorded in the database.

3.1.2 Data preprocessing:

The available data was divided into training set and test data. 80% of the data was used for the training set. Since the maximum number of responses were from people of age group 15-25 only 127 responses were taken into consideration. 100 users were taken for the purpose of training. User ratings of 100 users for different food items were created such that each user has at least 20 ratings.



10	1	4	42
11	1	2	121
12	1	2	137
13	1	5	201
14	1	3	23
15	1	4	3
16	1	2	88
17	1	3	182
18	1	1	172
19	1	5	222
20	1	5	18
21	2	5	224
22	2	1	118
23	2	5	228
24	2	3	8
25	2	4	155
26	2	1	212

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 collaborative techniques 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 to recommend items to the user based on the rating of other users having similar preferences. 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.

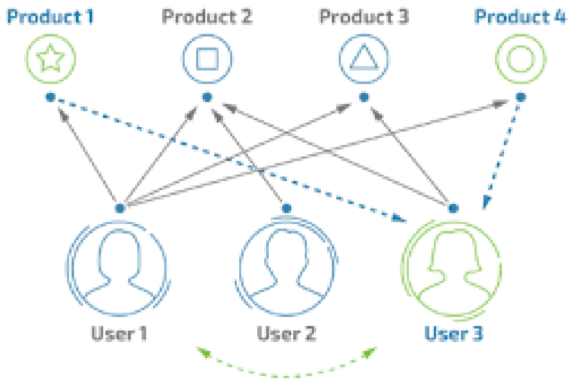


Figure 5- User-based collaborative filtering

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

∑xy = sum of products of paired scores

∑x = sum of x scores

∑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 scores between the users such that the users in descending order.
4. Produce weighted score that ranks the users by taking the multiplying their ratings of different foods with the similarity score.

B. 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.

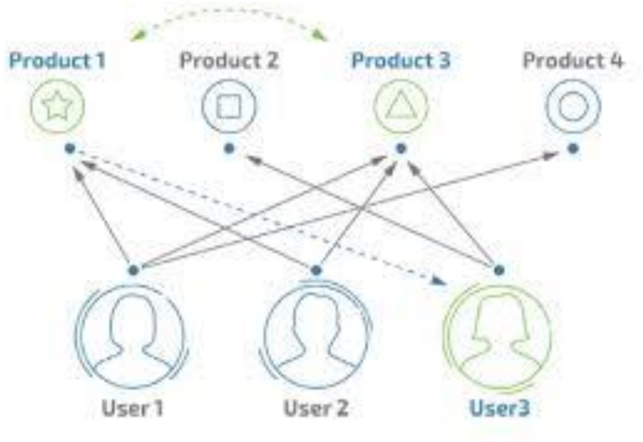


Figure 6- Item-based collaborative filtering

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 different users similarity score.

3.1.4 Validation of model

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

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

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

Where,

tp = true positive

fn = false positive

fp = false positive

3.2 System Design

3.2.1 Class diagram

There are five main classes used in the application. The user class consists of all the users registered in the application.

It contains a list of user_id, name, username, email address and food preference of each user. The user needs to enter the details while registering to the application. When a user tries to log into the application, it is checked if the username and password match the value in the database and if the value matches then the user is allowed to log into the system.

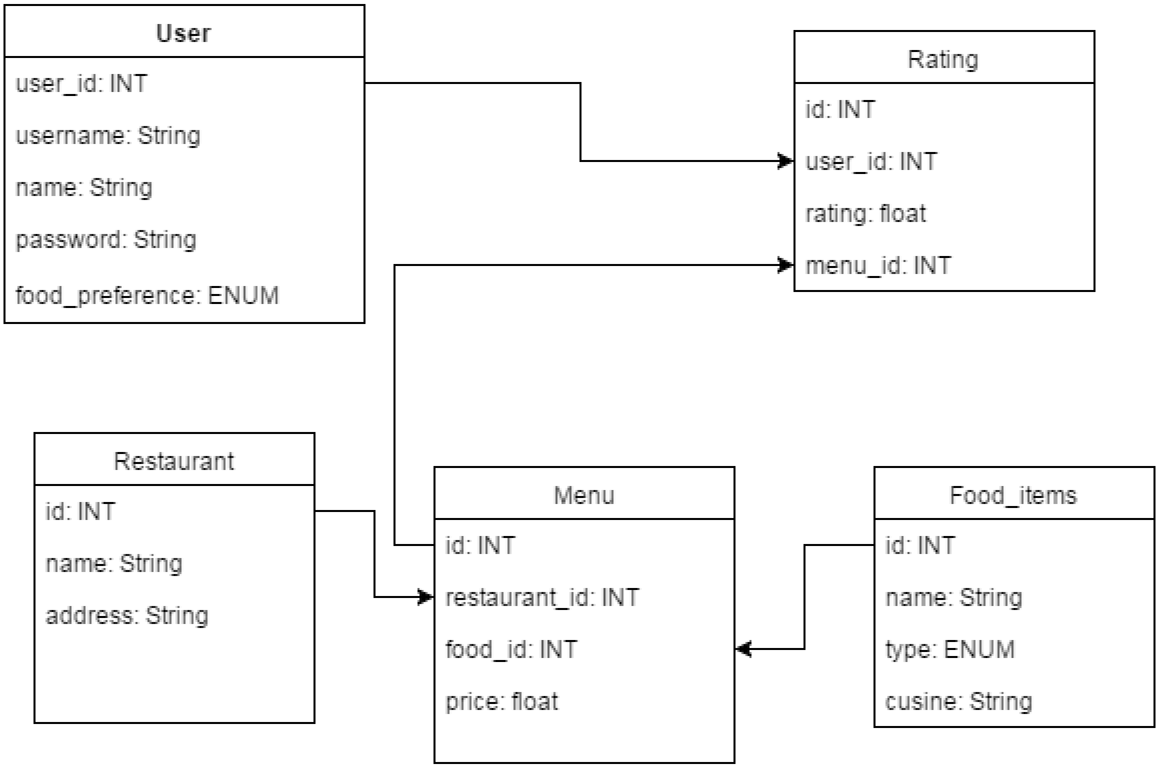


Figure 7- Class diagram of application

The Restaurant class contains the name of all the restaurants with their address and phone number. The address of the restaurant helps the user to find the location of the restaurant.

The class Food_items consists a list of unique food items along with its details, such as name, food type and cuisine it belongs.

The class Menu provides the menu of each restaurant in the database. The class references to the Restaurant class using restaurant_id and to Food_items class using food_id.

The Rating class consists of all the ratings provided by the users to different food items of different restaurants.

3.2.2 Sequence diagram

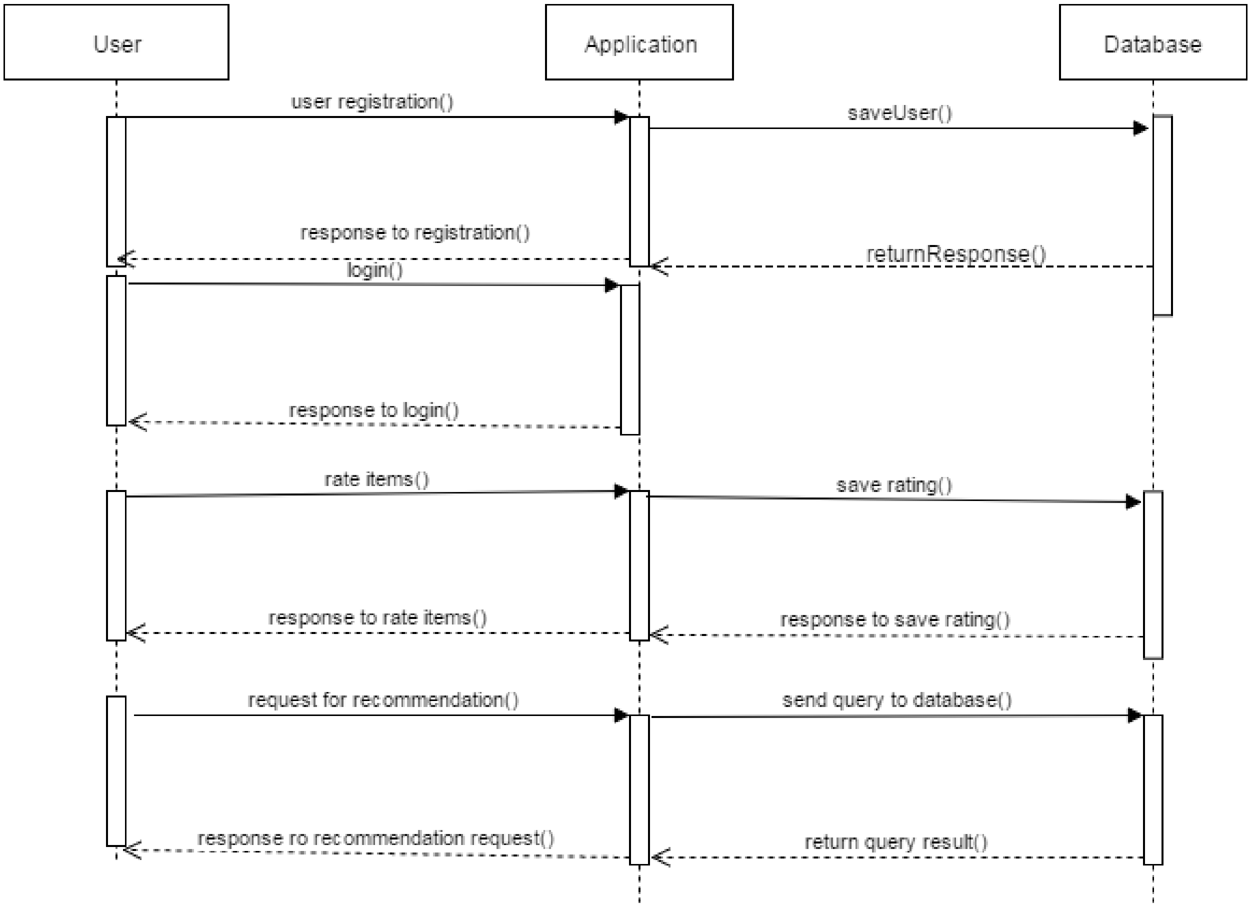


Figure 8- Sequence diagram of the application

The user first needs to register to the application. When the user fills in the registration form, the details are validated by the application and saved in the database. Then, the user successfully registers to the application

While the user tries to log in to the application with its user credentials, the application checks if the credentials are valid or not. If the credentials are valid, the user can log into the system, whereas if the credentials are incorrect, the user is redirected to the same login page.

When the user rates an item, the application first checks if the user has already rated the specific item. If yes, the user is not allowed to rate the item, else the value is saved in the database.

When the user asks for a recommendation, the application first checks if the user has selected user based filtering or item based application. If the user has asked for user-based filtering, then the similarity between the users are calculated and the result is displayed.

Whereas, if the user has selected item-based filtering, the application checks if the number of ratings by the user is greater than 20, if yes, the recommendation is made and the result is displayed, else the application sends an error message.

3.2.3 State diagram

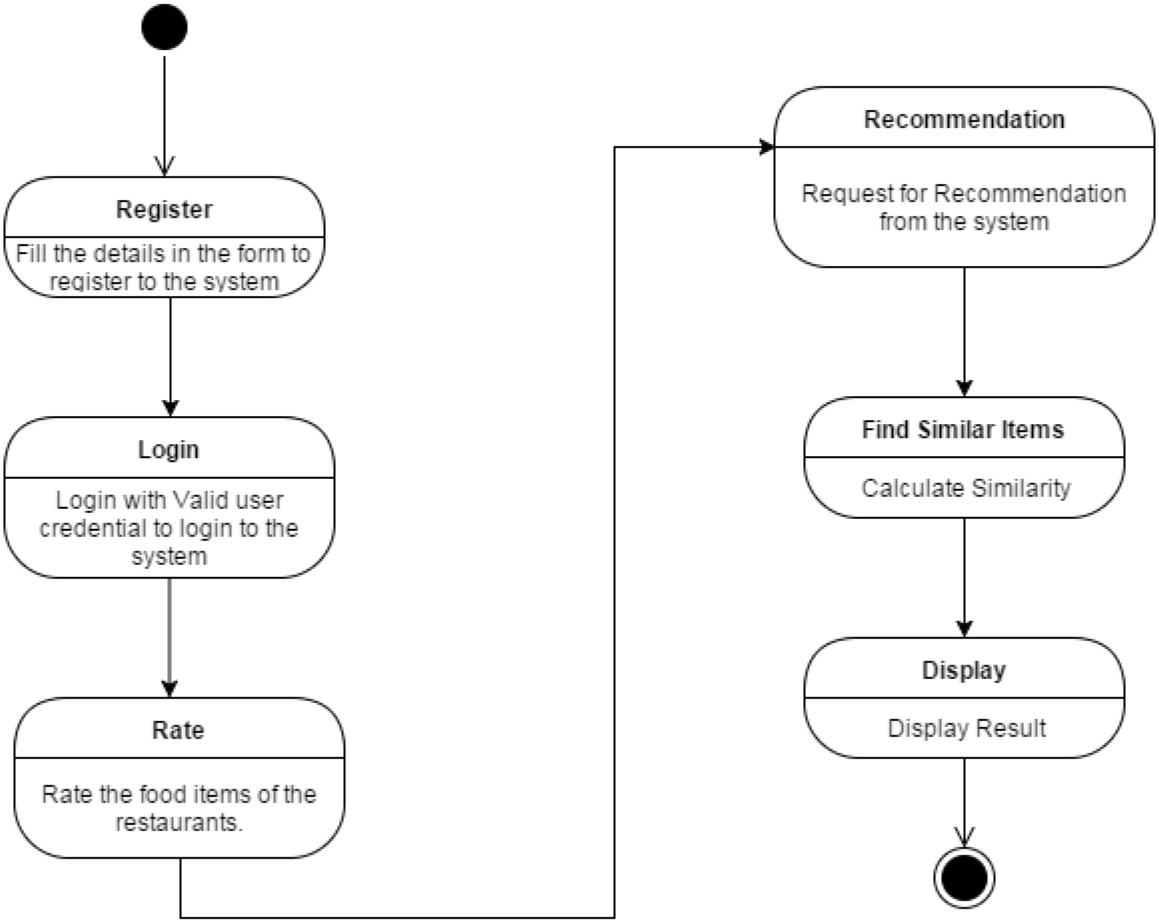

Register
Fill the details in the form to register to the system
Login
Login with Valid user credential to login to the system
Rate
Rate the food items of the restaurants.
Recommendation
Request for Recommendation from the system
Find Similar Items
Calculate Similarity
Display
Display Result

Figure 9- State diagram

The Figure 9 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 of different restaurants. The ratings of the users are saved into the database.

The user can also choose to get recommendations. When the user searches for the recommendation, 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 Flask framework. It uses python programming language in back-end and JavaScript in front-end. MySQL was used as DBMS for the application. The algorithms were implemented in python. JavaScript was used for validation.

Similarly, 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 between the 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 items
si={}
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 0
if 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/den
return r
```

4.2 Testing

The system was validated using Precision and Recall Model. The test date were used to test the system and the application obtained 71.42% precision and 62.5% Recall.

Table 3- Precision and recall

	Predicted Negative	Predicted Positive
Negative Cases	TN = 10	FP = 15
Positive Cased	FN = 20	TP = 25

Precision = $\frac{25}{25+10} =$ 71.45%

Recall = $\frac{25}{25+15} =$ 62.5%

CHAPTER 5: MAINTENANCE AND SUPPORT

The application will be maintained and updated over the period of time and necessary support will be provided to adapt the system to the future needs. Some of the strategies are:

5.1 Corrective Maintenance

As application could be sold or deployed for public use. There could be unresolved issues and if a user complains about it, the maintenance has to be done.

5.2 Adaptive Maintenance

The data in the application does not include all the restaurants in the valley, thus the data needs to be updated in future.

CHAPTER 6: CONCLUSION AND RECOMMENDATION

6.1. Conclusion

The project Restaurant Recommendation System was successfully completed by using User based and Item-based collaborative filtering. The data set were collected from survey which was preprocessed on the basis attributes. The data were then used to model the system.

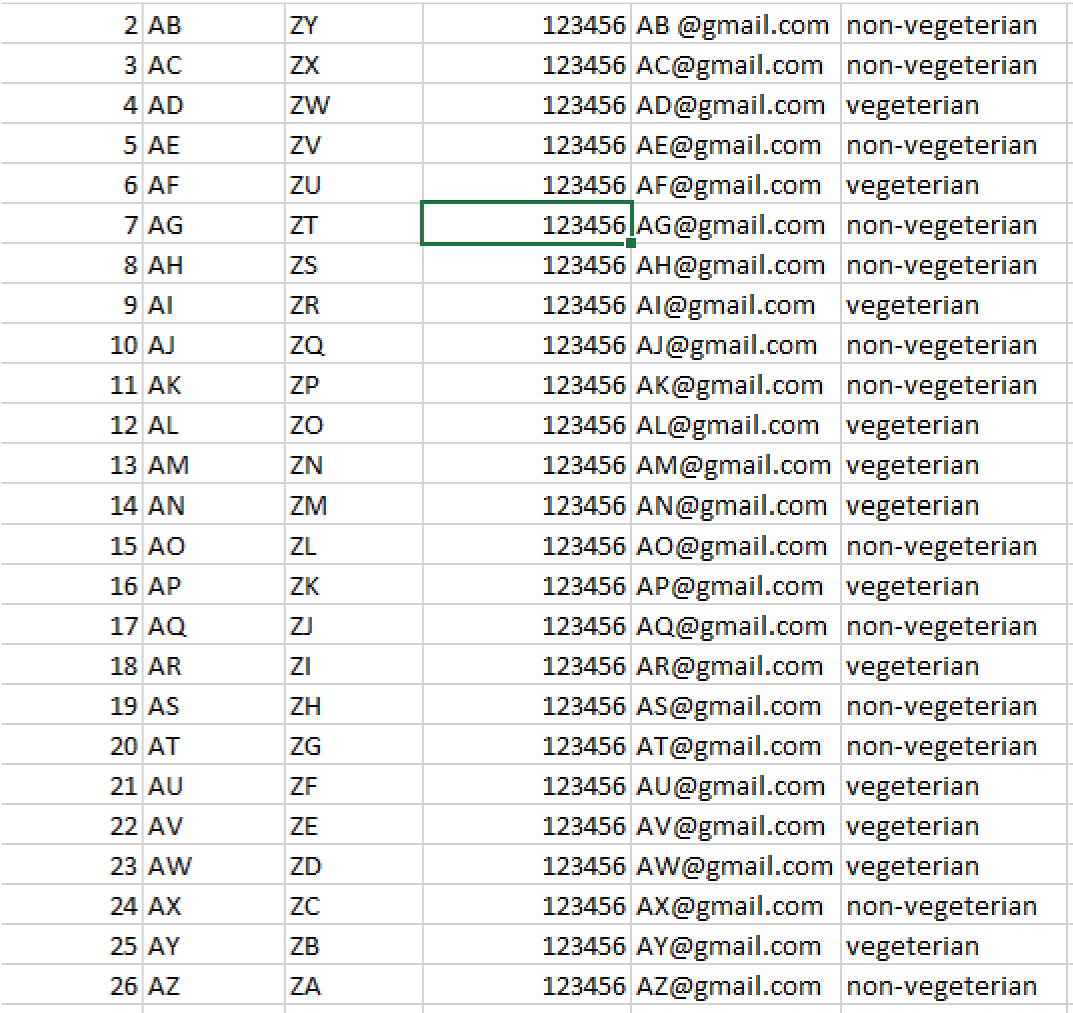
Successful implementation of was done by creating a web application, with 71.42% precision and 62.5% recall.

6.2. 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 restaurants in the valley. Furthermore, since the use of mobile phones is huge, the application will be more effective if built in mobile platform.

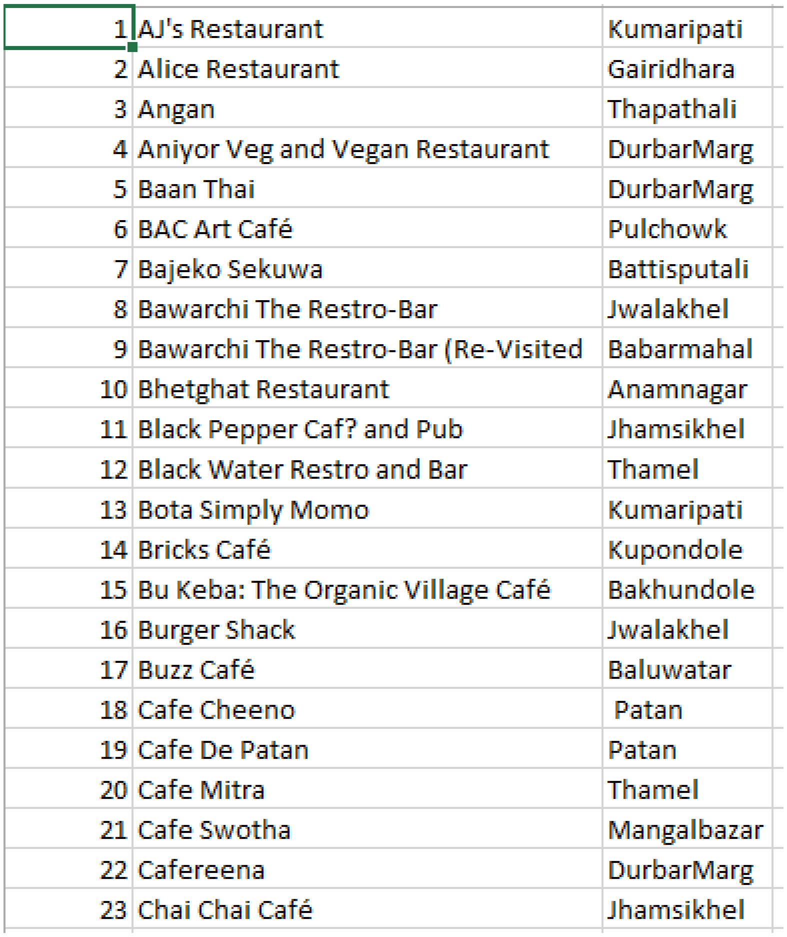
APPENDIX I

Sample data of user table:



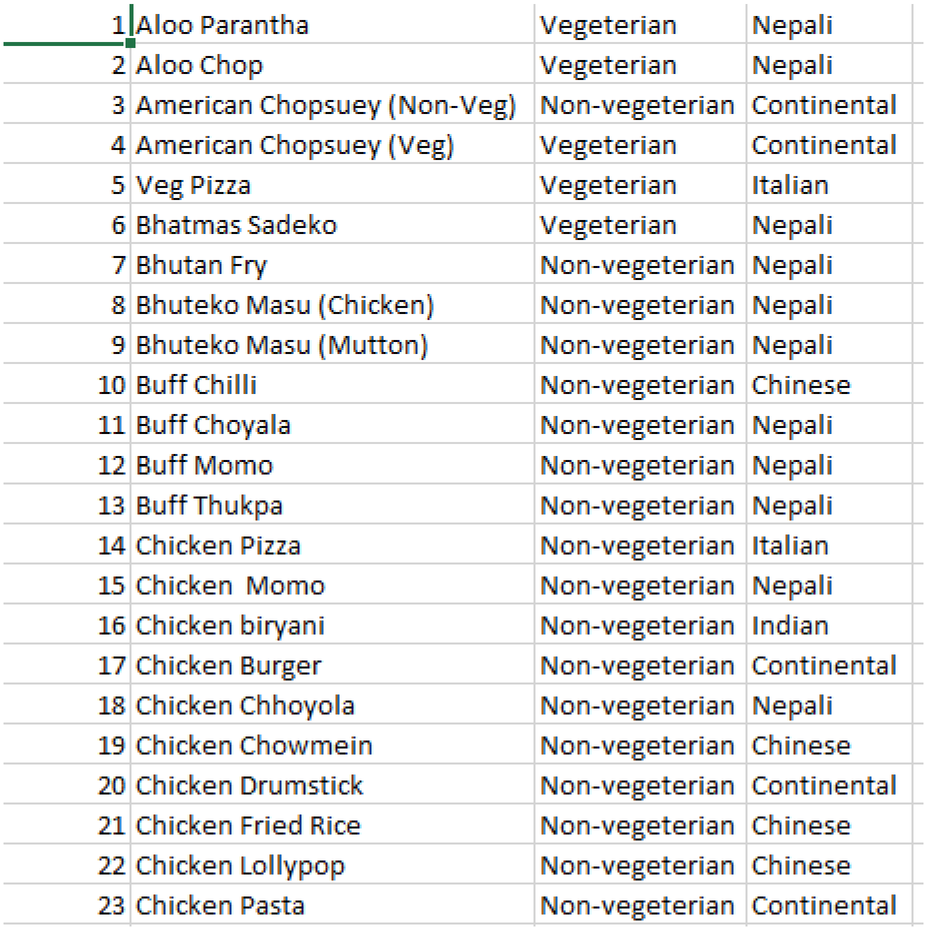
2	AB	ZY	123456	AB @gmail.com	non-vegeterian
3	AC	ZX	123456	AC@gmail.com	non-vegeterian
4	AD	ZW	123456	AD@gmail.com	vegeterian
5	AE	ZV	123456	AE@gmail.com	non-vegeterian
6	AF	ZU	123456	AF@gmail.com	vegeterian
7	AG	ZT	123456	AG@gmail.com	non-vegeterian
8	AH	ZS	123456	AH@gmail.com	non-vegeterian
9	AI	ZR	123456	AI@gmail.com	vegeterian
10	AJ	ZQ	123456	AJ@gmail.com	non-vegeterian
11	AK	ZP	123456	AK@gmail.com	non-vegeterian
12	AL	ZO	123456	AL@gmail.com	vegeterian
13	AM	ZN	123456	AM@gmail.com	vegeterian
14	AN	ZM	123456	AN@gmail.com	vegeterian
15	AO	ZL	123456	AO@gmail.com	non-vegeterian
16	AP	ZK	123456	AP@gmail.com	vegeterian
17	AQ	ZJ	123456	AQ@gmail.com	non-vegeterian
18	AR	ZI	123456	AR@gmail.com	vegeterian
19	AS	ZH	123456	AS@gmail.com	non-vegeterian
20	AT	ZG	123456	AT@gmail.com	non-vegeterian
21	AU	ZF	123456	AU@gmail.com	vegeterian
22	AV	ZE	123456	AV@gmail.com	vegeterian
23	AW	ZD	123456	AW@gmail.com	vegeterian
24	AX	ZC	123456	AX@gmail.com	non-vegeterian
25	AY	ZB	123456	AY@gmail.com	vegeterian
26	AZ	ZA	123456	AZ@gmail.com	non-vegeterian

Sample data of restaurant table:



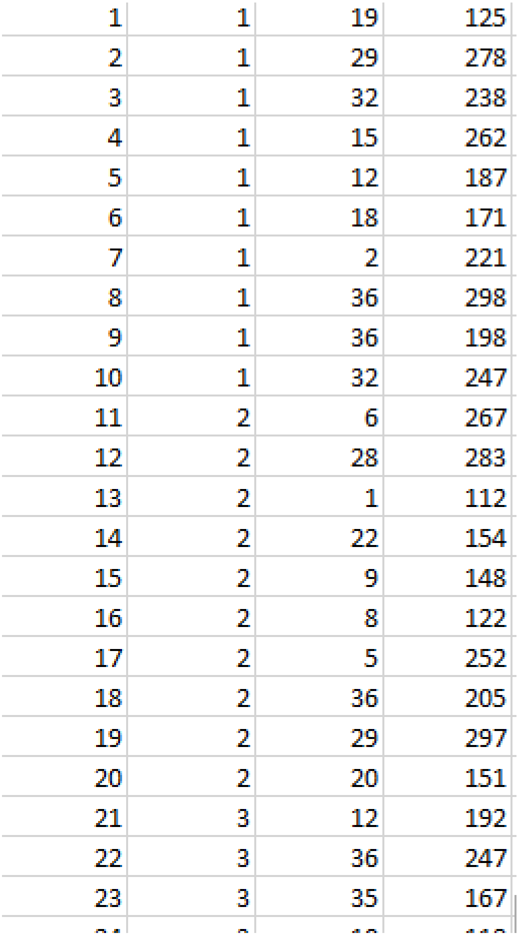
1	AJ's Restaurant	Kumaripati
2	Alice Restaurant	Gairidhara
3	Angan	Thapathali
4	Aniyor Veg and Vegan Restaurant	DurbarMarg
5	Baan Thai	DurbarMarg
6	BAC Art Café	Pulchowk
7	Bajeko Sekuwa	Battisputali
8	Bawarchi The Restro-Bar	Jwalakhel
9	Bawarchi The Restro-Bar (Re-Visited	Babarmahal
10	Bhetghat Restaurant	Anamnagar
11	Black Pepper Caf? and Pub	Jhamsikhel
12	Black Water Restro and Bar	Thamel
13	Bota Simply Momo	Kumaripati
14	Bricks Café	Kupondole
15	Bu Keba: The Organic Village Café	Bakhundole
16	Burger Shack	Jwalakhel
17	Buzz Café	Baluwatar
18	Cafe Cheeno	Patan
19	Cafe De Patan	Patan
20	Cafe Mitra	Thamel
21	Cafe Swotha	Mangalbazar
22	Cafereena	DurbarMarg
23	Chai Chai Café	Jhamsikhel

Sample data of food_items table



1	Aloo Parantha	Vegeterian	Nepali
2	Aloo Chop	Vegeterian	Nepali
3	American Chopsuey (Non-Veg)	Non-vegeterian	Continental
4	American Chopsuey (Veg)	Vegeterian	Continental
5	Veg Pizza	Vegeterian	Italian
6	Bhatmas Sadeko	Vegeterian	Nepali
7	Bhutan Fry	Non-vegeterian	Nepali
8	Bhuteko Masu (Chicken)	Non-vegeterian	Nepali
9	Bhuteko Masu (Mutton)	Non-vegeterian	Nepali
10	Buff Chilli	Non-vegeterian	Chinese
11	Buff Choyala	Non-vegeterian	Nepali
12	Buff Momo	Non-vegeterian	Nepali
13	Buff Thukpa	Non-vegeterian	Nepali
14	Chicken Pizza	Non-vegeterian	Italian
15	Chicken Momo	Non-vegeterian	Nepali
16	Chicken biryani	Non-vegeterian	Indian
17	Chicken Burger	Non-vegeterian	Continental
18	Chicken Chhoyola	Non-vegeterian	Nepali
19	Chicken Chowmein	Non-vegeterian	Chinese
20	Chicken Drumstick	Non-vegeterian	Continental
21	Chicken Fried Rice	Non-vegeterian	Chinese
22	Chicken Lollypop	Non-vegeterian	Chinese
23	Chicken Pasta	Non-vegeterian	Continental

Sample data of menu table



1	1	19	125
2	1	29	278
3	1	32	238
4	1	15	262
5	1	12	187
6	1	18	171
7	1	2	221
8	1	36	298
9	1	36	198
10	1	32	247
11	2	6	267
12	2	28	283
13	2	1	112
14	2	22	154
15	2	9	148
16	2	8	122
17	2	5	252
18	2	36	205
19	2	29	297
20	2	20	151
21	3	12	192
22	3	36	247
23	3	35	167

Appendix II

Survey form used for data collection

Name:

Age:

Below 15 years □

15- 25 years □

25-35 years □

Above 35 years □

REFERENCES

Asanov, D. (2011). *Algorithms and Methods in Recommender System.* Berlin Institute of Technology.

Ekstrand, M. D. (2010). ollaborative Filtering RecommenderSystems.

Jong- Hun Kim, U.-G. K.-H. (2006). *Content-Based Filtering for Music Recommendation Based on Ubiquitos Computing.* Incheon, Korea: Department of Computer Science anf Engineering Inha University.

Larry Page, S. B. (1994). *The Anatomy of a Large-Scale Hypertextual Web Search Engine.* Stanford University.

Linden, G. (2003). Amazon.com Recommendations. Item-to-Item Collaborative Filtering. 1-5.

Melville, P. (2010). Recommender Systems. *Recommender Systems*, 1.

Xiaoyuan Su, T. M. (2009). *A Survey of Collaborative Filtering Techniques.* Florida: Department of Computer Science and Engineering, Florida Atlantic University.