**Project Based Learning-II (AY 2021-22)**

**Batch-G2 Sem-II**

**Date- 14th May, 2022**

A picture containing shape

Description automatically generated

Group members (Roll no. and Name):

**Divya Lambhate (21249)**

**Vedant Madhe (21250)**

**Manas Sewatkar (21253)**

**Siddhesh Pande (21260)**

**Ajay Patil (21261)**

Lab Teacher Name: **Prof. P. Jambhulkar**

Sign:

Date:

Title of project

**“DVSAM: Restaurant Recommender System”**

**CERTIFICATE**

This is to certify that the work incorporated in the report entitled “**DVSAM: Restaurant Recommender System**” is carried out by Mr. Manas Chandrashekhar Sewatkar (Roll No-21253), who is part of a group of students for Project under the subject **Project Based Learning** during A.Y. **2021-22**.

Such material has not been submitted to any other University/ Institute for any financial support. The literature related to the problem investigated has been appropriately cited and duly acknowledged wherever facilities and suggestions have been availed of.

**Date: 12/05/2022 Name & Sign of Project Guide**

**Place: PUNE Prof. P. J. Jambhulkar**

**Prof. Rutuja Kulkarni Dr. G. V. Kale**

**Name & Sign of PBL Coordinator Name & Sign of Head of Department**

**ACKNOWLEDGEMENT**

I would like to express my special thanks of gratitude to my teacher Prof. Parag Jambhulkar and our principal Dr. R. Sreemathy who gave me the golden opportunity to do this project named “DVSAM:Restaurant Recommender System” . It helped me in doing a lot of Research and I came to know about a lot of things related to this topic.

Finally, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

**Place: Pune Name of Student (In capital) & Sign**

**MANAS CHANDRASHEKHAR SEWATKAR**

**DVSAM: Restaurant Recommender System**

# Introduction:

Recommendation systems are a subclass of information filtering systems that seek to predict the "rating" or "preference" a user would give to an item. Several fields like entertainment, music and social media are currently exploiting the potential of machine learning recommender systems that help grow their respective businesses. Some commonly recognized forms of recommender systems are playlist generators, product recommenders for online stores as well as online dating. Recommenders for exploring research articles, collaborators and financial services have also been developed.

Recommender Systems are broadly classified into two types:

1. **Collaborative Filtering:**

This method is purely based on the past interactions between customers and targeted products. The historical data is stored and scraped from the internet to explore the likings as well as disliking of different users.

This method lacks in considering present trends and culture as it solely depends on historical data and nothing else. However, it can recommend more personal recommendations as it analyses a customer’s individual preferences.

1. **Content-based Filtering:**

This method of recommendation generation employs additional information about the customer as well as the product. For better predictions and analysis, the algorithm takes into consideration information like date of birth, sex, region, linguistic preference, marital status, etc.

# Motivation:

Food is the basic requirement for any individual across all ages. It represents values, traditions, and culture of various segments of the population. However, the working class of people who move out of their hometowns find it difficult to track their own culture as well as explore other traditions and customs.

A restaurant recommender system helps users to satisfy their culinary needs and explore new cuisines and restaurants. It also helps restaurants to gain fame and grow as individual businesses. This type of recommender systems can also help individuals who travel a lot find restaurants and diners according to their tastes and preferences.

This project uses a dataset scraped and compiled by the popular food delivery system Zomato reviewing and crowd-sourcing platform Yelp and exploits data generated by millions of users living in and near the city of Pune.

# Objective:

The objective of this project is to provide quality restaurant recommendations for individuals across ages, culture, locality, and economic status. To measure the quality of recommendations we have used data from students in PICT to rate the recommendations from our model.

It is also proposed to help restaurants gain recognition via our website and uncover hidden gems in and around Pune. In our society, there are people who like to explore restaurants and diners and post their likings on social media platforms like Yelp. Our algorithms can scrape the data from these platforms and process it to form our own database to provide quality results for the users. In turn, restaurants also gain followers via social media influencers.

# Scope of Project and Intended Audience:

Our project exploits crowd-sourced data to generate recommendations. It is therefore limited to areas in and near the city of Pune. The project can be used by residents of Pune as well as visitors of Pune to get quality recommendations from our website.

**Overall Description**

# Functional Requirements:

The following are the functional requirements for accessing our website:

1. For computers or laptops:
   1. Internet connectivity.
   2. Web browser like Google Chrome, Microsoft Edge, etc.
   3. Intel i3 8th Gen Processor.
   4. 4GB RAM.
2. For mobile devices:
   1. Internet connectivity.
   2. Web browser.
3. For hosting website:
   1. Heroku application and account.
   2. Heroku command line interface
   3. Local Git Repository.
   4. Python libraries of Django for Back-end
   5. HTML, CSS and Bootstrap for Front-end.

# Non-Functional Requirements:

The following are the non-functional requirements we have met while creating this project:

1. Security: Django offers tokens to make sure that data cannot be tampered with while sending it to and from the backend and frontend sections of the web application.
2. Speed: We have used optimized Sci-Kit Learn and Pandas libraries and used concepts of OOP to improve the reaction speed of our web application.
3. Portability: Our web application has been created considering individuals accessing the website from numerous devices and has responsive features to improve user experience.

# Operating Environment:

We have used the following tech-stack for our project:

1. Back-end: Django
2. Front-end: HTML, CSS, Bootstrap
3. Machine Learning: Python, Pandas, Sci-Kit Learn
4. Database: SQLite (in-built)
5. Intel i7 10th Gen Processor
6. Windows 10 Operating System.

# Flowchart:

Diagram

Description automatically generated

# Implementation Details:

1. *Cleaning:*

Before implementing any machine learning algorithm, the data gathered by Zomato had to be cleaned in order to explore it.



Fig 1.1: Uncleaned Zomato Data

During cleaning, the pseudo-numeric values were converted to numeric values and unrelated columns were dropped.



Fig 1.2: Cleaned data

1. *Creating user profile to restaurant relation data:*

To implement collaborative filtering, a relational dataset is required which maps users to their rated restaurants. This data was collected via google forms from 500 students of PICT. The students filled the restaurant names from the list of restaurants in the Zomato dataset and assigned ratings as per their choice.

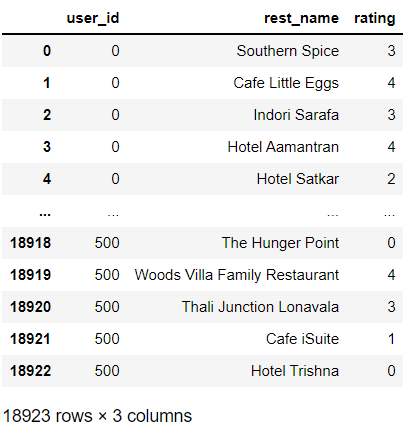


Fig 2.1: Mapped data

1. *Pivoting dataset to generate matrix:*

The data set was pivoted to generate a matrix of restaurant name to restaurant name to implement cosine similarity model.

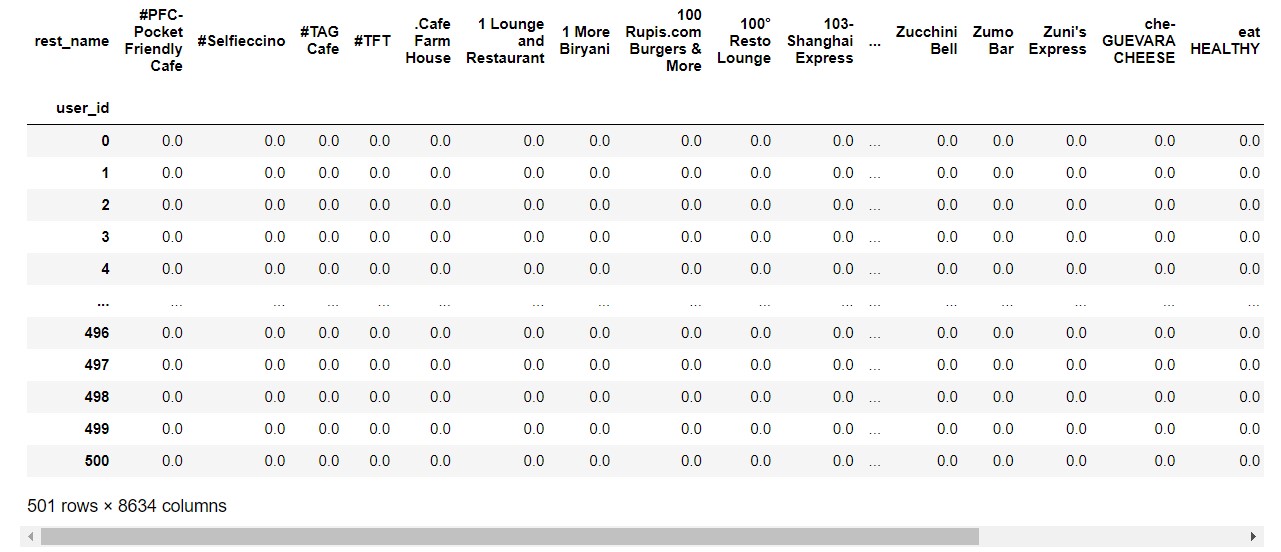


Fig 3.1: Mapping of restaurant users to all the restaurants (Pivot Table)

1. *Standardizing Data:*

The pivot table was standardized in order to eliminate harsh ratings, non-rated restaurants and un-visited restaurants in the table.

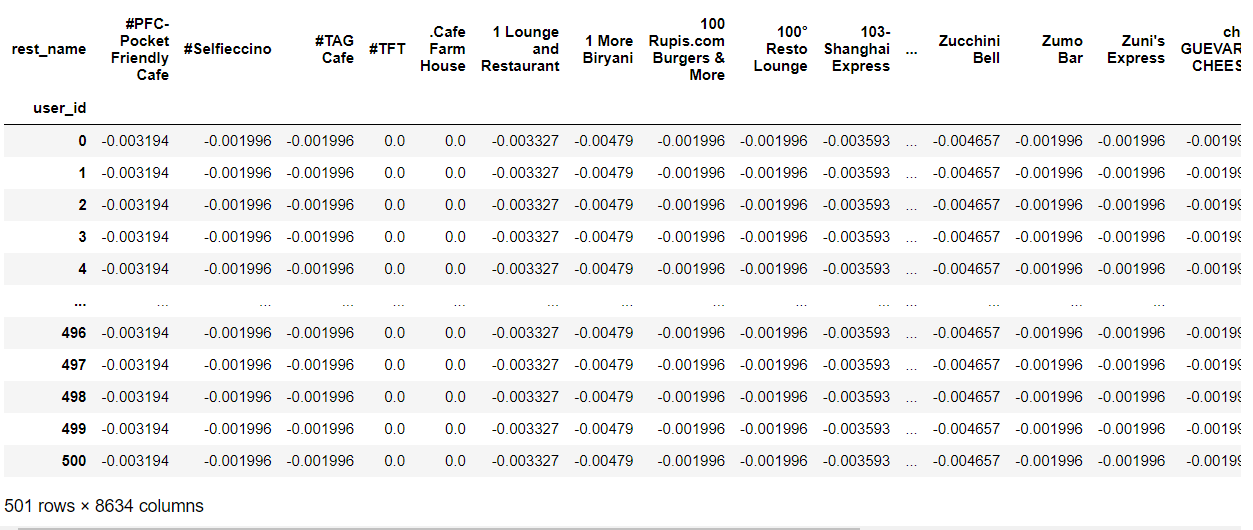


Fig 4.1: Standardized form of user-to-restaurant mapping.

1. *Creating Similarity Table:*

The pivot table was used to create a similarity table using cosine similarity module of Sci-Kit Learn library. Using this we received a 2-D array which contained similarity between restaurants on the basis of the users (students) who rated them.

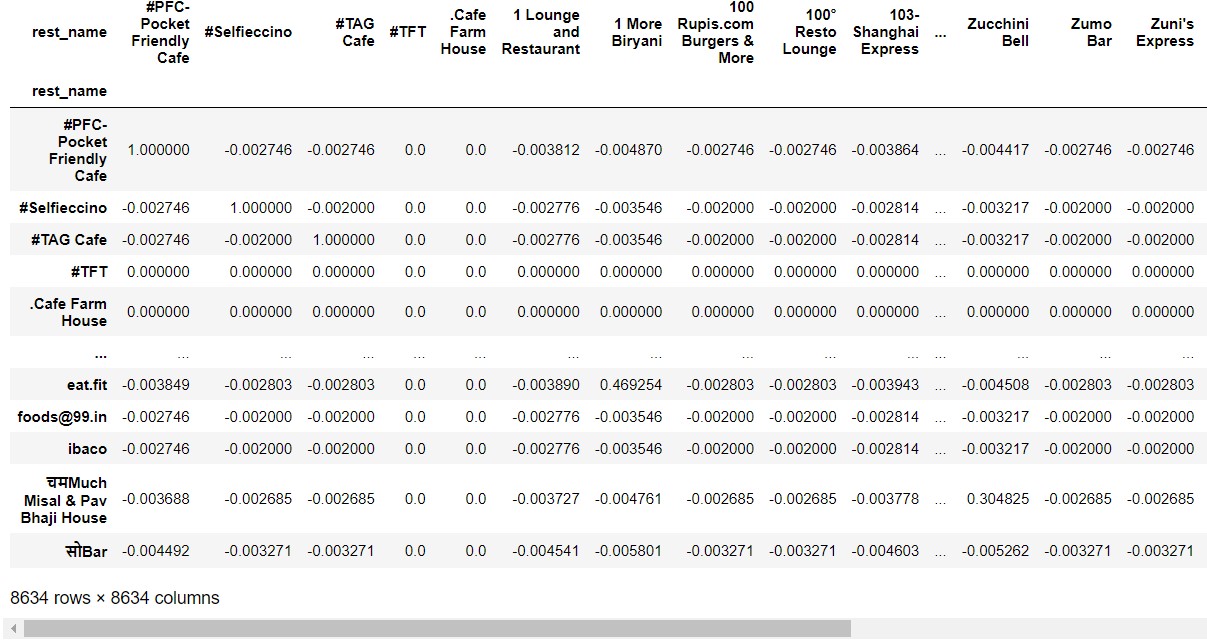


Fig 5.1: Similarity Table

1. *Getting similar restaurants:*

To get similar restaurants, similarity scores of the given restaurant were calculated based on its user rating. Sorting the restaurants in the descending order, the top 10 restaurants were considered as the best recommendations. On doing this for all the restaurants that a particular user has rated, the compiled list of recommendations is sorted again to give overall recommendations.

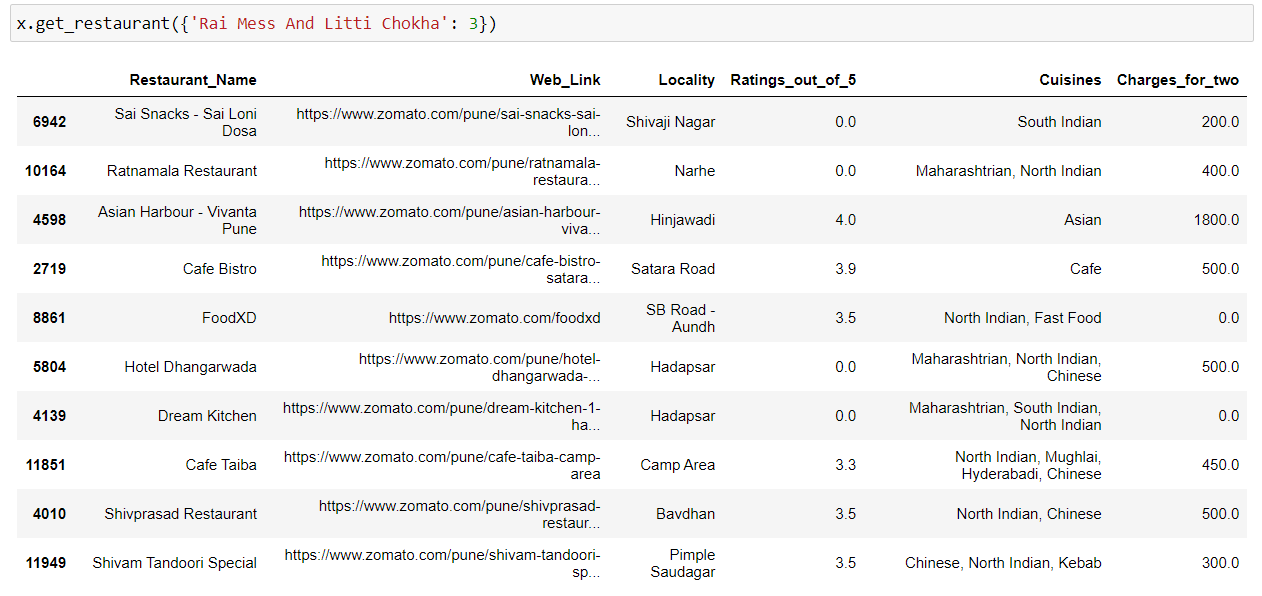


Fig 6.1: Compiled list of similar restaurants for Rai Mess and Litti Chokha.

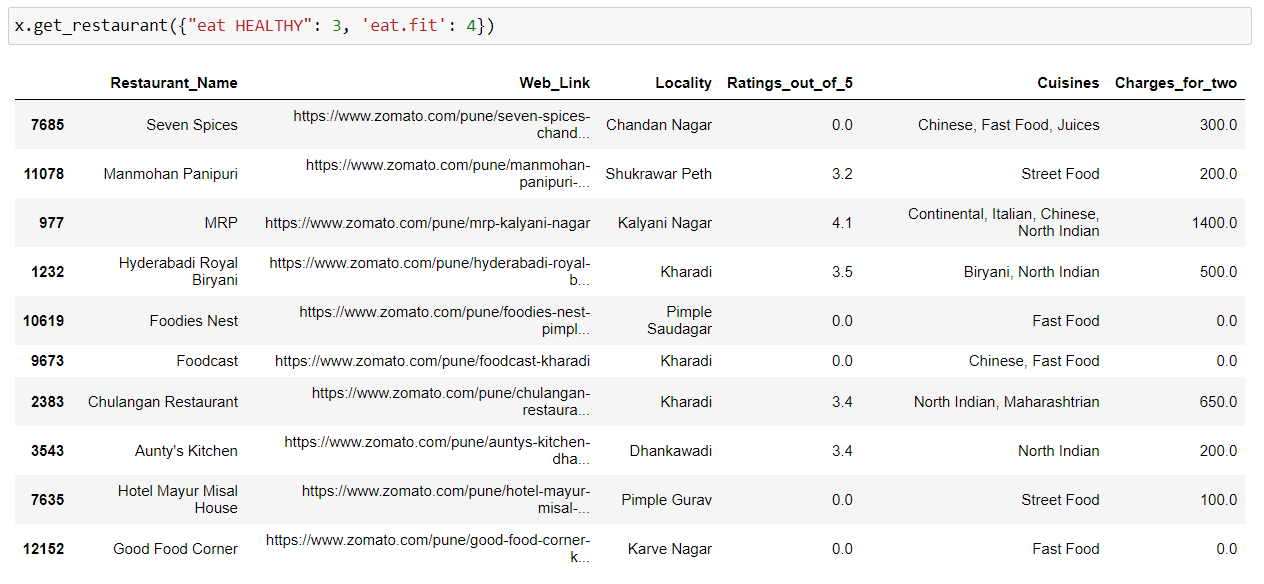


Fig 6.4: Compiled list of similar restaurants for multiple restaurants.

1. *Displaying data on webpages:*

The data from machine learning model was used in Django back-end framework to show the data in graphical format on the users’ screen.

* 1. Django User model stores data in the form of 10 variables to store the restaurant names and ratings of the last five restaurants the user has rated.
  2. This data is sent to the machine learning model in the form of dictionaries.
  3. The data is retrieved in the form of Django data frame which is passed to the front-end via *json records* encoded as a context dictionary.
  4. Django also allowed the use of python statements in HTML pages to loop and print all the data in the json record on a HTML page.
  5. The HTML pages were designed using CSS code and Bootstrap framework.

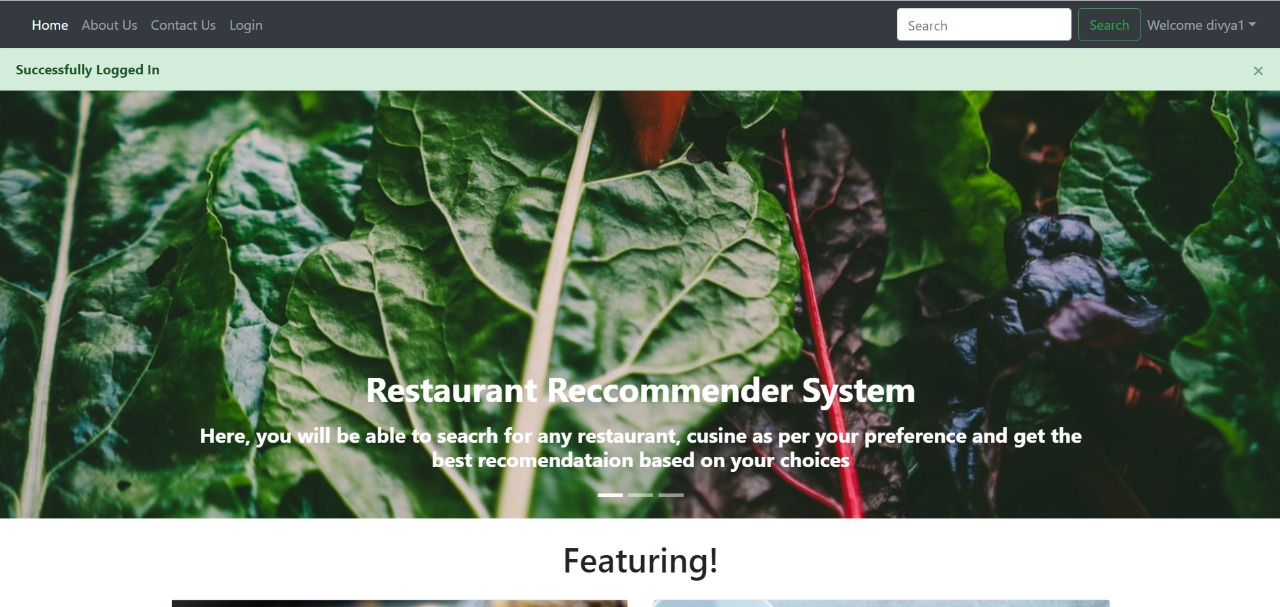


Fig 7.1: Home page of Website

Graphical user interface

Description automatically generated with medium confidence

Fig 7.2: Login page for Users.

Graphical user interface, application

Description automatically generated

Fig 7.3: Sign Up page for Users.

**Conclusion**

The objective of this study was to recommend quality recommendations to users by using their history and ratings by using a web-based interface.

The collaborative filtering approach to recommendation systems also ensures that user privacy is maintained as similar users are recommended restaurants that they might like based on crowd-sourced models. The only data used by the recommender system being the rating that the users have given to the restaurants. Unlike content-based filtering, the complex logic to assign weights to different features of a particular restaurant is abstracted in collaborative filtering and therefore makes it more efficient for application and prediction of restaurants.

The future of this project has a lot of potential to be a successful platform to search for restaurants. The restaurants recommended will have better quality if more data can be obtained by making the system learn from real-time data. To implement such a system would be a Herculean task but worth the time and money spent on it.

**References**

# For Machine Learning:

* Machine Learning by Andrew Ng: <https://youtube.com/playlist?list=PLLssT5z_DsK-h9vYZkQkYNWcItqhlRJLN>
* Python for Data Science and Machine Learning Bootcamp: <https://www.udemy.com/share/101WaU3@1xt0aFh3JARiLIQv_pK1BRwpw72UzwmS_QW7ruWwyt6RNG6Hxjc9IjwRYk2nIjvJLg==/>

# For Web-Development:

* MDN documentation: <https://developer.mozilla.org/>
* Django Documentation: <https://docs.djangoproject.com/en/4.0/>
* Django Tutorial by Corey Schafer: <https://youtube.com/playlist?list=PL-osiE80TeTtoQCKZ03TU5fNfx2UY6U4p>
* Bootstrap Documentation: <https://getbootstrap.com/>