Food Recommendation System by using Principal Component Analysis

SUBMITTED TO

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

In the partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOG

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

THOTA MAHESH RAJU
20NG1A0559

Under the Esteemed Guidance of

Ch Phani Kumar

Assistant Professor

AUTONOMOUS

(Approved by AICTE and JNTUK, Kakinada)

(ON NH 16, TELAPROLU, NEAR GANNAVARAM - 521109)

2020-2024

USHA RAMA
COLLEGE OF ENGINEERING AND TECHNOLOGY

AUTONOMOUS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Affiliated of to JNTU Kakinada, Approved by A.I.C.T.E, New Delhi)

TELAPROLU, UNGUTURU MANDAL, KRISHNA DISTRICT-521109

2023-2024

CERTIFICATE

This is to certify that this project entitled **"Food Recommendation system by using Principal Component Analysis"** is the bonafide work of **THOTA MAHESH RAJU (20NG1A0559)** who carried out the work under my supervision, and submitted in partial fulfilment of the requirements for the award of the degree in Bachelor of Technology in Computer Science & Engineering, during the academic year 2023-24.

Project Guide	**Head of the Department**
Ch. Phani Kumar	**Dr. K P N V SATYA SREE**
Assistant Professor	

Signature of External Examiner

DECLARATION

I Here by declare that the project entitled **“Food Recommendation system by using Principle Component Analysis”** is the work done by us during the academic year 2023-2024 and is submitted in partial fulfilment of the requirements for the award of degree of **Bachelor of technology** in **COMPUTER SCIENCE AND ENGINEERING** from **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA**.

BY

THOTA MAHESH RAJU
(20NG1A0559)

ACKNOWLEDGEMENT

We express our sincere thanks where it is due

I am pleased to acknowledge our sincere thanks to our Honorable Chairman **SRI. S.RAMABRAHMAM GARU** for the support and advice which is given and for providing sufficient resources.

I am extremely thankful to **Dr. K RAJASEKHARA RAO**, Director of USHA RAMA COLLEGE OF ENGINEERING AND TECHNOLOGY, TELAPROLU for giving a golden opportunity to our education and project work.

I wish to avail this opportunity to express our thanks to **Dr. G V K S V PRASAD**, Principal, URCE for his continuous support and giving valuable suggestions during the entire period of the project work.

I take this opportunity to express our gratitude to **Dr. K P N V SATYA SREE**, Head of the Department Professor in **Computer Science and Engineering** for her valuable support and motivation at each and every point in successful completion of the project.

My sincere thanks to our guide **Ch. Phani Kumar assistant Professor** in the Department of Computer Science and Engineering for motivating us to make our project successful. We grateful for her guidance and suggestions.

I am also place our floral gratitude to all other teaching staff and lab technicians for their constant support and advice throughout the project.

BY
THOTA MAHESH RAJU
(20NG1A0559)

FOOD RECOMMENDATION SYSTEM BY USING PRINCIPAL COMPONENT ANALYSIS

ABSTRACT

ABSTRACT

Food recommendation systems have gained immense popularity in recent years due to the convenience they offer in helping users discover and order their preferred dishes from a vast array of restaurants. In this research paper, we propose an innovative approach to enhance the efficiency and accuracy of food recommendations by integrating Principal Component Analysis (PCA) for dimensionality reduction, using Swiggy's extensive dataset as a case study.

The primary objective of this study is to improve user experience and increase the relevance of food recommendations. We achieve this by reducing the dimensionality of the dataset, making it more manageable and computationally efficient, while preserving the essential features for generating personalized recommendations. The Swiggy dataset used in this research encompasses a wide range of information, including restaurant ratings, city locations, menu item costs, restaurant names, cuisine types, and links to place orders.

Keywords: Machine Learning, PCA, Dimensionality reduction, eigen values and eigen values, covariance Matrix, Python flask, Gmail SMTP server, WT Forms, FPDF.

TABLE OF CONTENTS

CHAPTER - 1

INTRODUCTION

1. INTRODUCTION

The rapid expansion of the food delivery industry has revolutionized the way people access and enjoy their favorite meals. In this age of convenience, online food delivery platforms have seamlessly woven themselves into the fabric of urban life. Among these platforms, Swiggy stands out as a prominent player, offering an extensive network of restaurants and a diverse array of culinary choices. As competition in this sector continues to intensify, the demand for more sophisticated and personalized food recommendation systems has become increasingly evident.

Traditional food recommendation systems have often relied on methods such as collaborative filtering, content-based filtering, or hybrid approaches to suggest dishes or restaurants to users. While these approaches have shown promise, they are not without their limitations. Collaborative filtering heavily relies on user interaction data and may struggle to provide meaningful recommendations for new users with limited history. Content-based filtering, on the other hand, often depends on extensive feature engineering, which can be labor-intensive and may not fully capture the nuanced preferences of users.

In response to these challenges, this research paper introduces an innovative approach to enhance the efficiency and accuracy of the food recommendation system by incorporating Principal Component Analysis (PCA) for dimensionality reduction. PCA is a widely used technique in machine learning and data analysis, helping identify the most informative features while reducing the computational complexity of large datasets.

The dataset used in this study is sourced from Swiggy, one of India's leading food delivery platforms. It comprises a wealth of information, including restaurant ratings, city locations, menu item costs, restaurant names, cuisine types, and links to place orders. Leveraging this rich dataset, we aim to develop a recommendation system that not only streamlines the recommendation process but also ensures that the suggestions are tailored to the user's preferences and constraints.

The primary motivation behind this research is to enhance the user experience on food delivery platforms. We recognize that users have diverse tastes, budgets, and geographic preferences when it comes to food. To address this, our proposed system allows users to input their specific criteria. They can select their desired city, set minimum rating thresholds for restaurants, and establish maximum cost limits for menu items. This empowers users to have control over the recommendations they receive, ensuring they align with their unique preferences.

The core innovation of this research lies in the integration of PCA for dimensionality reduction within the food recommendation system. This enables us to reduce the dataset's dimensionality while preserving crucial information for generating personalized recommendations. The end result is a streamlined and efficient recommendation process that caters to the individual needs of users.

The key contribution of this research is the creation of a result table that contains essential information for users to make informed decisions. This table includes restaurant ratings, city locations, menu item costs, restaurant names, cuisine types, and convenient links for placing orders. Through thoughtful consideration of user-defined criteria and the application of PCA, our system empowers users to discover new dining experiences that align precisely with their preferences and constraints.

`

1.1 LITERATURE SURVEY

Food recommendation systems have become a vital component of online food delivery platforms, enhancing user satisfaction and engagement. Over the years, researchers and practitioners have explored various techniques and approaches to improve the accuracy and effectiveness of these systems. In this literature survey, we review key studies and methodologies that have contributed to the development of food recommendation systems, with a focus on the integration of Principal Component Analysis (PCA) for dimensionality reduction.

[1]. Collaborative Filtering:
Collaborative filtering is a widely used technique in recommendation systems, relying on user-item interaction data to generate suggestions. Early collaborative filtering methods suffered from data sparsity issues and cold start problems, making them less effective for new users or items. However, advancements in collaborative filtering, such as matrix factorization and deep learning-based approaches, have shown promise in addressing these limitations.

[2]. Content-Based Filtering:
Content-based filtering systems leverage item attributes and user preferences to make recommendations. While this approach offers the advantage of explainability, it often requires extensive feature engineering. Researchers have explored natural language processing and machine learning techniques to extract relevant features from textual data, such as restaurant descriptions and user reviews.

[3]. Hybrid Recommendation Systems: Hybrid recommendation systems combine collaborative and content-based filtering methods to improve recommendation quality. These systems aim to harness the strengths of both approaches while mitigating their respective weaknesses.

Research in hybrid systems has focused on finding optimal ways to blend collaborative and content-based signals effectively.

[4]. Dimensionality Reduction Techniques:
Dimensionality reduction techniques, including PCA, have been applied in various recommendation system contexts. PCA, in particular, has shown promise in reducing the dimensionality of user-item interaction matrices while preserving essential information. Studies have explored the application of PCA in user and item embeddings to enhance the efficiency of collaborative filtering.

[5]. Personalization and User Preferences: Recognizing the importance of personalization in food recommendation, researchers have investigated methods to capture and incorporate user preferences and constraints. This includes allowing users to set location preferences, price ranges, dietary restrictions, and preferred cuisines. Personalized recommendation systems have gained traction as they enhance user satisfaction and engagement.

1.1.1 MACHINE LEARNING

A Machine Learning defined as “A computer program is said to learn from experience and from some tasks and some performance on, as measured by, improves with experience”. Machine Learning is combination of correlations and relationships, most machine learning algorithms in existence are concerned with finding and/or exploiting relationship between datasets. Once Machine Learning Algorithms can pinpoint on certain correlations, the model can either use these relationships to predict future observations or generalize the data to reveal interesting patterns.In Machine Learning there are various types of algorithms such as Regression, Linear Regression, Logistic Regression, Naive Bayes Classifier, Bayes theorem, KNN (K-Nearest Neighbour Classifier), Decision Tress, Entropy, ID3, SVM (Support Vector Machines), K-means Algorithm, Random Forest and etc.,

The name machine learning was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with datamining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning.

With in the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover" hidden in sights" through learning from historical relationships and trends in the data

Machine learning implementations are classified into three major categories, depending on the nature of the learning “signal” or “response” available to a learning system which are as follows:

Supervised learning:

When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of Supervised learning. This approach is indeed similar to human learning under the supervisionof a teacher. The teacher provides good examples for the student to memorize, and the student derives general rules from these specific examples.

Unsupervised learning:

When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of un-correlated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms. As a kind of learning, it resembles the methods humans use to figure out that certain objects or events are from the same class, such as by observing the degree of similarity between objects. Some recommendation systems that you find on the web in the form of marketing automation are based on this type of learning.

Reinforcement learning:

When you present the algorithm with examples that lack labels, as in unsupervised learning. However, you can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which is connected to applications for which the algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences. In the human world, it is just like learning by trial and error.Errors help you learn because they have a penalty added (cost, loss of time, regret, pain, and so on), teaching you that a certain course of action is less likely to succeed than others.

In this case, an application presents the algorithm with examples of specific situations, such as having the gamer stuck in a maze while avoiding an enemy. The application lets the algorithm know the outcome of actions it takes, and learning occurs while trying to avoid what it discovers to be dangerous and to pursue survival. You can have a look at how the company Google Deep Mind has created a reinforcement learning program that plays old Atari's video 3 games. When watching the video, notice how the program is initially clumsy and unskilled but steadily improves with training until it becomes a champion.

Semi-supervised learning:

Where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing. Supervised Learning the majority of practical machine learning uses supervised learning. Supervised learning is where you have input variables (x) and an output variable (Y)and you use an algorithm to learn the mapping function from the input to the output.

$$Y = f(X)$$

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data. It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

Types of Supervised Learning:

Classification:

It is a Supervised Learning task where output is having defined labels (discretevalue). For example, in above Figure A, Output – Purchased has defined labels i.e., 0 or 1; 1 means the customer will purchase and 0 means that customer won't purchase. The goal here is to predict discrete values belonging to a particular class and evaluate on the basis of accuracy. It can be either binary or multi class classification. In binary classification, model predicts either 0 or 1; yes or no but in case of multi class classification, model predicts more than one class. Example: Gmail classifies mails in more than one classes like social, promotions,updates, forum.

Regression:

It is a Supervised Learning task where output is having continuous value. Examplein above Figure B, Output – Wind Speed is not having any discrete value but is continuous in the particular range. The goal here is to predict a value as much closer to actual output value asour model can and then evaluation is done by calculating error value. The smaller the errorthe greater the accuracy of our regression model.

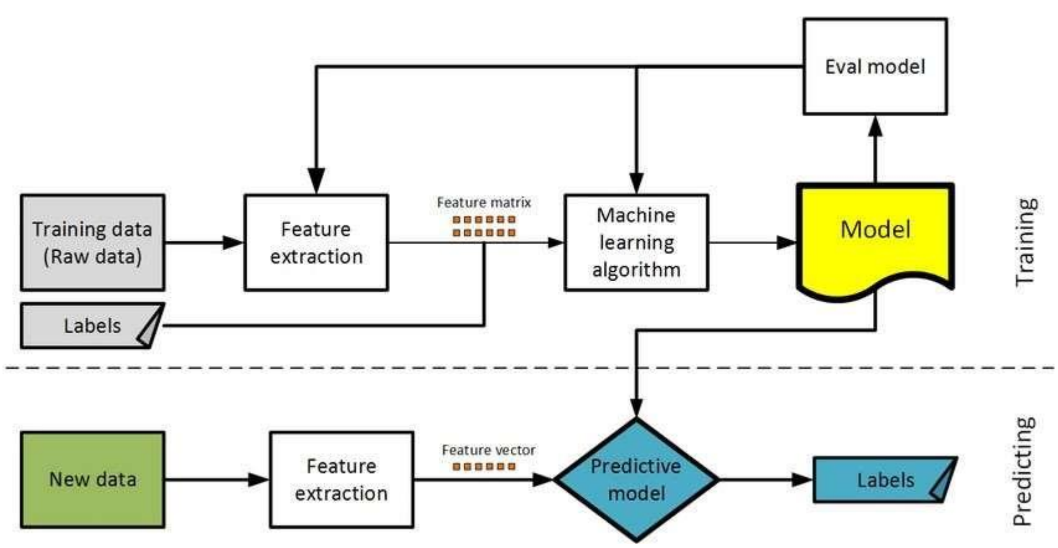


Fig: 1.1.1. FLOW CHART OF SUPERVISED LEARNING ALGORITHM

Classification:

Data mining is the process of extracting knowledge-able information from huge amounts of data. It is an integration of multiple disciplines such as statistics, machine learning, neural networks and pattern recognition. Data mining extracts biomedical and health care knowledge for clinical decision making and generates scientific hypotheses from large medical data.

Association rule mining and classification are two major techniques of data mining. Association rule mining is an unsupervised learning method for discovering interesting patterns and their association in large data bases.

Classification is a supervised learning method used to find class labels for unknown samples. Classification is the task of assigning an object's tone of special predefined categories. It is pervasive problem that encompasses many applications.

Classification is designed as the task of learning a target function F that maps each attribute setA to one of the predefined class labels C. The target function is also known as classification model. A classification model is useful for mainly two purposes.

1) descriptive modelling.
2) Predictive modelling.

Classification is the process of recognizing, understanding, and grouping ideas and objects into pre-set categories or “sub-populations.” Using pre-categorized training datasets, machine learning programs use a variety of algorithms to classify future datasets into categories.

Classification algorithms in machine learning use input training data to predict the likelihood that subsequent data will fall into one of the predetermined categories. One of the most common uses of classification is filtering emails into “spam” or “non-spam.”

In short, classification is a form of “pattern recognition,” with classification algorithms applied to the training data to find the same pattern (similar words or sentiments, number sequences, etc.) in future sets of data.

Classification can be performed on structured or unstructured data. Classification is a technique where we categorize data into a given number of classes. The main goal of a classification problem is to identify the category/class to which a new data will fall under.

Few of the terminologies encountered in machine learning – classification:

Classifier: An algorithm that maps the input data to a specific category.

Classification model: A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.

Feature: A feature is an individual measurable property of a phenomenon being observed.

Binary Classification: Classification task with two possible outcomes. E.g., Gender classification (Male / Female).

Multi-class classification: Classification with more than two classes. In multi class classification each sample is assigned to one and only one target label. E.g., An animal can be cat or dog but not both at the same time.

Multi-label classification: Classification task where each sample is mapped to a set of target labels (more than one class). E.g., A news article can be about sports, a person, and location at the same time.

Applications of Classification Algorithms:

- Email spam classification
- Bank customers loan pay willingness prediction.
- Cancer tumour cells identification.
- Sentiment analysis
- Drug's classification
- Facial key points detection
- Pedestrians' detection in an automotive car driving.

1.1.1. FEATURES OF MACHINE LEARNING

- It is nothing but automating the Automation.
- Getting computers to program themselves.
- Writing Software is bottleneck.
- Machine leaning models involves machines learning from data without the help of humans or any kind of human intervention.
- Machine Learning is the science of making of making the computers learn and act like humans by feeding data and information without being explicitly programmed.

- Machine Learning is totally different from traditionally programming, here data and output is given to the computer and in return it gives us the program which provides solution to the various problems. Below is the figure

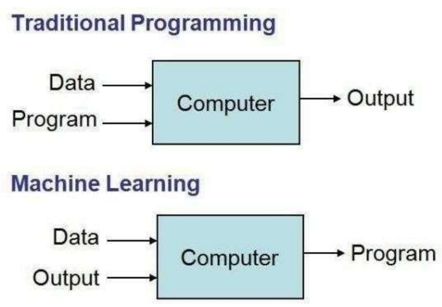
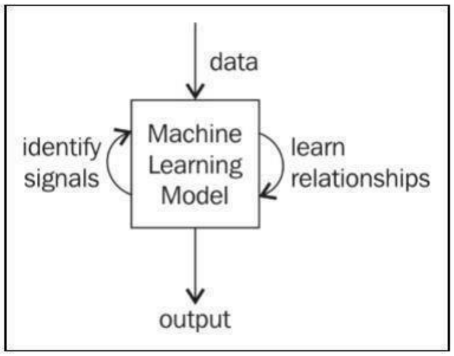


Fig: 1.1.1.1. TRADITIONAL PROGRAMMING VS MACHINE LEARNING

- Machine Learning is a combination of Algorithms, Datasets, and Programs.
- There are Many Algorithms in Machine Learning through which we will provide us the exactsolutionin predicting the disease of the patients.
- How Does Machine Learning Works?
- Solution to the above question is Machine learning works by taking in data, finding relationships within that data and then giving the output.



An overview of machine learning models

Fig: 1.1.1.2 MACHINE LEARNING MODEL

There are various applications in which machine learning is implemented such as Web search, computing biology, finance, e-commerce, space exploration, robotics, social networks, debugging and much more.

1.1.3. EXISTING SYSTEM

The food recommendation system based on Principal Component Analysis (PCA) is designed to provide personalized food recommendations to users. The system operates by first gathering a comprehensive dataset that encompasses a wide range of food items, each described by various attributes like ingredients, nutritional content, user ratings, and potentially more. PCA, a dimensionality reduction technique, is then applied to this dataset. Its primary objective is to discover latent factors or underlying patterns within the food data. These latent factors could represent taste profiles, dietary categories, or other intrinsic features of the foods, which may not be immediately evident.

User interactions play a crucial role in this system. Users are required to input their preferences, dietary restrictions, and perhaps even past consumption history. This information is utilized to create user profiles that help tailor the recommendations. By aligning the user profiles with the latent factors extracted through PCA, the system can suggest food items that closely match the user's individual taste and dietary requirements. This personalization is a key feature, as it enhances the user experience and increases the likelihood of user satisfaction with the recommendations.

Moreover, the system is designed to be adaptive and iterative. It continuously collects feedback from users regarding their food choices and adjusts the recommendations accordingly. This feedback loop is essential for refining the system's accuracy and ensuring that it adapts to changing user preferences over time, the existing food recommendation system employs PCA to unveil hidden patterns in food data, integrates user profiles to provide personalized suggestions, and continually refines itsmrecommendations through user feedback. This combination of data analysis, personalization, and adaptation results in a more effective and user-centric food recommendation system.

PCA Drawbacks:

Linearity: PCA assumes that the relationships between variables are linear. It may not capture complex, non-linear relationships in the data.

Sensitivity to Scaling: PCA is sensitive to the scale of the variables. Rescaling or standardizing variables can significantly impact the results**.**

Interpretability: While PCA reduces dimensionality, the resulting principal components may not have clear or intuitive interpretations, making it challenging to explain the meaning of the components.

Information Loss: PCA reduces the dimensionality by discarding less important components. This can lead to some loss of information, which may be critical in certain applications.

Outliers: PCA can be sensitive to outliers in the data, and a few extreme data points can heavily influence the principal components.

Data Distribution: PCA assumes that the data is normally distributed, which may not hold for all types of data, leading to suboptimal results.

Overemphasis on Variance: PCA prioritizes capturing variance in the data, which may not always align with the goals of a particular analysis, such as preserving specific features or relationships.

1.1.4. PROPOSED SYSTEM

We came up with the proposed system, where the users can find the restaurant nearby and food recommendations. A food recommendation system built on Principal Component Analysis (PCA) involves several key components. Firstly, data acquisition is crucial, where a wide range of users' food preferences and dietary restrictions are collected. This dataset should be diverse and comprehensive to ensure the system can cater to a broad spectrum of tastes and needs. PCA is then applied to this dataset, reducing its dimensionality by identifying underlying taste components. These components could represent attributes like spiciness, sweetness, or texture, allowing for a nuanced understanding of users' taste profiles.

User preferences are mapped onto these taste components, enabling the system to effectively recommend food items that align with a user's unique combination of taste preferences. Collaborative filtering techniques can be employed to suggest meals based on similar users' profiles, ensuring a personalized experience. Moreover, the system should take into account any dietary restrictions or allergies users might have, offering recommendations that are safe and suitable.

Continuous learning and adaptation are crucial for a food recommendation system. User feedback, gathered through interactions with the system, can be used to refine the PCA model and improve the accuracy of recommendations over time. The system should also be equipped with user-friendly web and mobile interfaces, making it accessible and convenient for users to discover new dishes and flavors, this food recommendation system based on PCA aims to provide users with personalized and diverse meal suggestions while considering their unique taste profiles and dietary constraints. It's a dynamic system that evolves through user feedback, with the ultimate goal of enhancing the overall culinary experience for its users.

CHAPTER-2

AIM & SCOPE

2. AIM & SCOPE

The aim of this project is to design and develop an innovative Food Recommendation System that leverages Principal Component Analysis (PCA) to enhance the accuracy and personalization of food recommendations for users. By applying PCA to user preferences and dietary data, the system will extract the most significant features and reduce the dimensionality of the dataset, leading to more efficient and effective food recommendations.

This project will encompass the creation of a robust recommendation system that combines user profiles, food preferences, and dietary restrictions to generate personalized food recommendations. The system will involve data collection from users, including their historical food choices, and apply PCA to model the data effectively. Additionally, it will consider ethical considerations, such as user privacy and data transparency. The resulting system will not only provide tailored food suggestions but also offer insights into the influence of PCA on recommendation accuracy. This project aims to benefit individuals seeking personalized dietary guidance, food enthusiasts, and the food service industry by improving user satisfaction and promoting healthier eating habits.

2.1. FEASIBILITY STUDY:

2.1.1.Technical Feasibility:

Project Description:

The project aims to implement Principal Component Analysis (PCA) to
reduce the dimensionality of a large dataset for improved data analysis and visualization

Technical Requirements:

Access to appropriate software for PCA analysis (e.g., Python with libraries like NumPy etc)
Sufficient computing resources to handle large datasets.
Knowledge and expertise in machine learning and data analysis techniques.

Technical Feasibility Assessment:

We have access to the necessary software and hardware.
The project team has the required technical skills.
Availability of relevant datasets for analysis.

2.1.2 Economic Feasibility:

Costs:

Software licenses and tools.
Hardware upgrades or cloud computing resources.
Personnel and training costs

Revenue/Value:

The project's value lies in improved data analysis, decision-making, and potentially saving costs

Economic Feasibility Assessment:

Estimated project costs are within the allocated budget.
Potential value and benefits outweigh the costs.

2.1.3. Operational Feasibility

Implementation Process:

Data collection and preprocessing.
PCA model development and testing.
Integration into existing data analysis workflows.

Operational Feasibility Assessment:

The project timeline aligns with academic schedules.
The project team is adequately staffed and trained.
Integration with existing systems is feasible.

2.1.4. Marketing Feasibility:

Market Demand:

Evaluate the need for PCA in various fields such as finance, healthcare, and engineering.
Identify potential users or beneficiaries of PCA result

Competitive Analysis:

Assess the presence of competing PCA solutions or alternatives.

Marketing Feasibility Assessment:

There is a demonstrated demand for PCA in research and industry.
The project can cater to a niche market with limited competition.

2.1.5. Ethical Feasibility:

Data Ethics:

Ensure that data used in PCA analysis adheres to privacy and ethical guidelines.
Prevent bias in data analysis and decision-making.

Transparency and Accountability:

Maintain transparency in the PCA process.
Provide clear documentation of the analysis and results.

Ethical Feasibility Assessment:

The project will follow ethical data handling and analysis practices.
Promote transparency and accountability in all stages of the project.

2.2. SYSTEM REQUIREMENT SPECIFICATION

A Software Requirements Specification (SRS) – a requirements specification for a software system– is a complete description of the behaviour of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software.In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification is a structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, isresponsible for analysing the business needs of their clients and stakeholders to help identify business problems and propose solutions.

2.2.1. FUNCTIONAL REQUIREMENTS:

A Functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define whata system is supposed to accomplish. Behavioural requirements describing all cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements(also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements,security, or reliability).

1. User Registration and Profiles: Users should be able to create accounts and set up profiles, including information about their food preferences, dietary restrictions, and allergies.

2. Data Collection: The system must collect and maintain a substantial dataset of food items, recipes, and user preferences.

3. Principal Component Analysis (PCA): Implement PCA to reduce the dimensionality of the food preference dataset, identifying key taste components.

4. Recommendation Engine: Develop an algorithm that uses PCA results to recommend food items based on a user’s taste profile and dietary constraints.

5. User Interaction: Provide user-friendly interfaces (web and mobile) for users to interact with the system, input preferences, and receive recommendations.

6. Collaborative Filtering: Incorporate collaborative filtering to recommend meals based on the preferences of users with similar taste profiles.

2.2.2. NON-FUNCTIONAL REQUIREMENTS:

Non-functional Requirements refer to the constraints or restrictions on the system. They may relate to emergent system properties such as reliability, response time and storeoccupancy or the selection of language, platform, implementation techniques and tools.

The non-functional requirements can be built on the basis of needs of the user, budget constraints, organization policies and etc.

1. Performance:

The system should respond quickly to user requests for food recommendations, with minimal latency. The system should be capable of handling a growing user base and dataset without a significant drop in performance. Efficient PCA calculations to ensure timely recommendations without delays.

2. Security:

Ensure the confidentiality and privacy of user data, adhering to data protection regulations.Implement secure user authentication and authorization mechanisms to protect user accounts and personal information. Use encryption protocols to secure data transmission between users and the system.

3. Usability:

Design a user interface that is intuitive, easy to navigate, and visually appealing for both web and mobile platforms. Ensure the system is accessible to users with disabilities, following accessibility standards.

4. Accuracy and Adaptability:

Continuously improve the accuracy of food recommendations by refining the PCA model based on user feedback. The system should adapt to evolving food trends and user preferences.

5. Reliability:

Ensure the system is available 24/7 with minimal downtime. Implement regular data backups and a robust recovery plan to handle system failures or data loss. Adhere to relevant legal and industry-specific regulations, especially regarding food safety and data privacy.

6. Interoperability:

The system should be able to integrate with external APIs or services, such as food delivery platforms or restaurant TV databases. Implement load balancing to evenly distribute user requests and maintain system stability during peak usage.

ACCESSIBILITY:

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. In our project people who have registered with the registration page can access their data with the help of login. User interfaces simple and efficient and easy to use

MAINTAINABILITY:

In software engineering, maintainability is the ease with which a software product can be modified in order to include new functionalities can be added in the project based on the user requirements just by adding the appropriate files to existing project using .net and programming languages. Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

SCALABILITY:

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users.

PORTABILITY:

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependent assemblies would have to be configured in such case.

VALIDATION:

It is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development life cycle. Software validation checks that the software product satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements, not as specification artifacts or as needs of those who will operate the software only but as the needs of all the stakeholders.

2.2.3. HARDWARE REQUIREMENTS

- System Processor : Intel i3 and above
- Hard Disk : 1 GB(min)
- RAM : 4GB(Min)

2.2.4. SOFTWARE REQUIREMENTS
2.2.5.

- Operating System : Windows
- Front-end : HTML, CSS, Java Script
- Back-end : Mysql Database
- Software Tools : Python(3.11.4) or visual studio code
- Packages : Pandas,Flask,Numpy,Sklearn,cv2,os,Flask,scikit-learn,Gmail SMTP server

CHAPTER - 3

CONCEPTS & METHODS

3. CONCEPTS & METHODS

3.1. PROBLEM DEFINITION

The project's core objective is to help the user in choosing the right and best restaurant related to their taste. In a world where the variety of food options is vast and diverse, individuals often face the daunting challenge of deciding what to eat. This project seeks to address the problem of culinary choice overload by creating a food recommendation system that harnesses the power of data analysis and PCA. The primary issue at hand is the need to assist users in discovering and selecting meals that align with their unique taste preferences and dietary requirements. This project aims to mitigate this problem by developing a system that can accurately understand, model, and recommend food items based on individual taste profiles, considering factors such as flavor, texture, and ingredient preferences.

Additionally, the system must overcome the challenge of handling dietary restrictions, allergies, and cultural preferences to ensure that recommendations are not only personalized but also safe and culturally relevant. To accomplish this, the system must collect and analyze a vast dataset of user food preferences, apply PCA to distill this data into meaningful taste components, and effectively match users with suitable meal choices. User engagement and feedback will be essential for refining the system over time, continuously improving its ability to suggest satisfying and diverse food options. The project’s ultimate goal is to simplify the food decision-making process, making it enjoyable and stress-free while promoting culinary exploration and diversity.

3.2. PROJECT DESCRIPTION

The project for a food recommendation system utilizing Principal Component Analysis (PCA) is a comprehensive and innovative solution designed to revolutionize the way people discover and enjoy food. At its core, the system aims to provide users with personalized food recommendations based on their unique taste preferences and dietary requirements. The project's scope begins with data collection, where a diverse dataset encompassing users' food preferences and restrictions is gathered. This dataset serves as the foundation for the PCA analysis.

Principal Component Analysis is the project's central technique, employed to reduce the dimensionality of the dataset and extract latent taste components. These components represent underlying factors such as flavor profiles, ingredient combinations, and culinary styles. By mapping users' preferences onto these components, the system gains a nuanced understanding of each user's distinct taste profile.

The system's recommendation engine is driven by collaborative filtering, which leverages these taste profiles to suggest food items that align with a user's unique combination of preferences. To enhance the user experience, the system takes into account dietary restrictions, allergies, and cultural preferences, ensuring that recommendations are not only personalized but also safe and culturally relevant.

Continuous learning and adaptation are fundamental to the project's success. It incorporates a feedback loop where user interactions are collected and used to fine-tune the PCA model and improve recommendation accuracy over time. The project also encompasses the development of user-friendly web and mobile interfaces, making it easy for users to explore and discover new culinary experiences.

In essence, the Food Recommendation System by Principal Component Analysis project is a dynamic and user-centric solution that aims to cater to individual tastes, promote culinary diversity, and enhance the overall food discovery and consumption experience. It combines cutting-edge data analysis with user feedback to offer a unique and evolving culinary journey for users, ultimately making mealtimes more enjoyable and personalized.

3.2.1. ALGORITHM PROPOSED

Principal Component Analysis (PCA)

As the number of features or dimensions in a dataset increases, the amount of data required to obtain a statistically significant result increases exponentially. This can lead to issues such as overfitting, increased computation time, and reduced accuracy of machine learning models this is known as the curse of dimensionality problems that arise while working with high-dimensional data.

As the number of dimensions increases, the number of possible combinations of features increases exponentially, which makes it computationally difficult to obtain a representative sample of the data and it becomes expensive to perform tasks such as clustering or classification because it becomes. Additionally, some machine learning algorithms can be sensitive to the number of dimensions, requiring more data to achieve the same level of accuracy as lower-dimensional data.

To address the curse of dimensionality, feature engineering techniques are used which include feature selection and feature extraction. Dimensionality reduction is a type of feature extraction technique that aims to reduce the number of input features while retaining as much of the original information as possible.

In this article, we will discuss one of the most popular dimensionality reduction techniques i.e. Principal Component Analysis(PCA).

What is Principal Component Analysis(PCA)?

Principal component analysis (PCA) technique was introduced by the mathematician **Karl Pearson** in 1901**.** It works on the condition that while the data in a higher dimensional space is mapped to data in a lower dimension space, the variance of the data in the lower dimensional space should be maximum.

- **Principal Component Analysis (PCA)** is a statistical procedure that uses an orthogonal transformation that converts a set of correlated variables to a set of uncorrelated variables. PCA is the most widely used tool in exploratory data analysis and in machine learning for predictive models. Moreover,

- Principal Component Analysis (PCA) is an unsupervised learning algorithm technique used to examine the interrelations among a set of variables. It is also known as a general factor analysis where regression determines a line of best fit.

- The main goal of Principal Component Analysis (PCA) is to reduce the dimensionality of a dataset while preserving the most important patterns or relationships between the variables without any prior knowledge of the target variables.

Principal Component Analysis (PCA) is used to reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables, retaining most of the sample's information, and useful for the regression and classification of data.

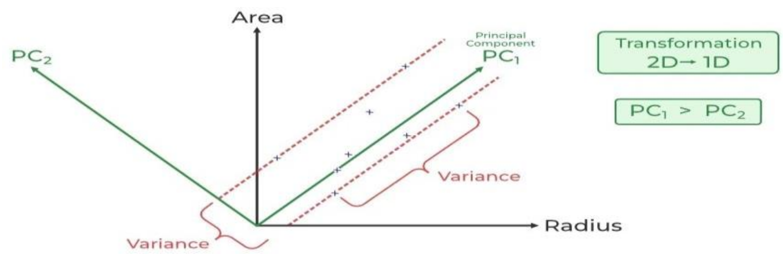


Fig:3.2.1. Principal component analysis

1) Principal Component Analysis (PCA) is a technique for dimensionality reduction that identifies a set of orthogonal axes, called principal components, that capture the maximum variance in the data. The principal components are linear combinations of the original variables in the dataset and are ordered in decreasing order of importance. The total variance captured by all the principal components is equal to the total variance in the original dataset.

2) The first principal component captures the most variation in the data, but the second principal component captures the maximum variance that is orthogonal to the first principal component, and so on.

3) Principal Component Analysis can be used for a variety of purposes, including data visualization, feature selection, and data compression. In data visualization, PCA can be used to plot high-dimensional data in two or three dimensions, making it easier to interpret. In feature selection, PCA can be used to identify the most important variables in a dataset. In data compression, PCA can be used to reduce the size of a dataset without losing important information.

4) In Principal Component Analysis, it is assumed that the information is carried in the variance of the features, that is, the higher the variation in a feature, the more information that features carries.

Overall, PCA is a powerful tool for data analysis and can help to simplify complex datasets, making them easier to understand and work with.

Step-By-Step Explanation of PCA (Principal Component Analysis)

3.2.1.1.Step 1: Standardization:

First, we need to standardize our dataset to ensure that each variable has a mean of 0 and a standard deviation of 1.

Z = \frac{X-\mu}{\sigma}

Here,

\mu is the mean of independent features

$\mu = \left \{ \mu_1, \mu_2, \cdots, \mu_m \right \}$

σ is the standard deviation of independent features

$\sigma = \left \{ \sigma_1, \sigma_2, \cdots, \sigma_m \right \}$

3.2.1.2. Step2: Covariance Matrix Computation

Covariance measures the strength of joint variability between two or more variables, indicating how much they change in relation to each other. To find the covariance we can use the formula:

$$Cov(x1,x2) = \frac{\sum_{i=1}^{n}(x1_i-\bar{x1})(x2_i-\bar{x2})}{n-1}$$

The value of covariance can be positive, negative, or zeros.

Positive: As the x1 increases x2 also increases.

Negative: As the x1 increases x2 also decreases.

Zeros: No direct relation

3.2.1.3. Step 3: Compute Eigenvalues and Eigenvectors of Covariance Matrix to Identify Principal Components

Let A be a square nXn matrix and X be a non-zero vector for which

$$AX = \lambda X$$

For some scalar values

λ . then

λ is known as the eigenvalue of matrix A and X is known as the eigenvector of matrix A for the corresponding eigenvalue.

It can also be written as :

$$\begin{aligned} AX-\lambda X &= 0 \\ (A-\lambda I)X &= 0 \end{aligned}$$

Where I am the identity matrix of the same shape as matrix A. And the above conditions will be true only if

$(A - \lambda I)$ will be non-invertible (i.e. singular matrix). That means,

$|A - \lambda I| = 0$

From the above equation, we can find the eigenvalues λ, and therefore corresponding eigenvector can be found using the equation

$AX = \lambda X$.

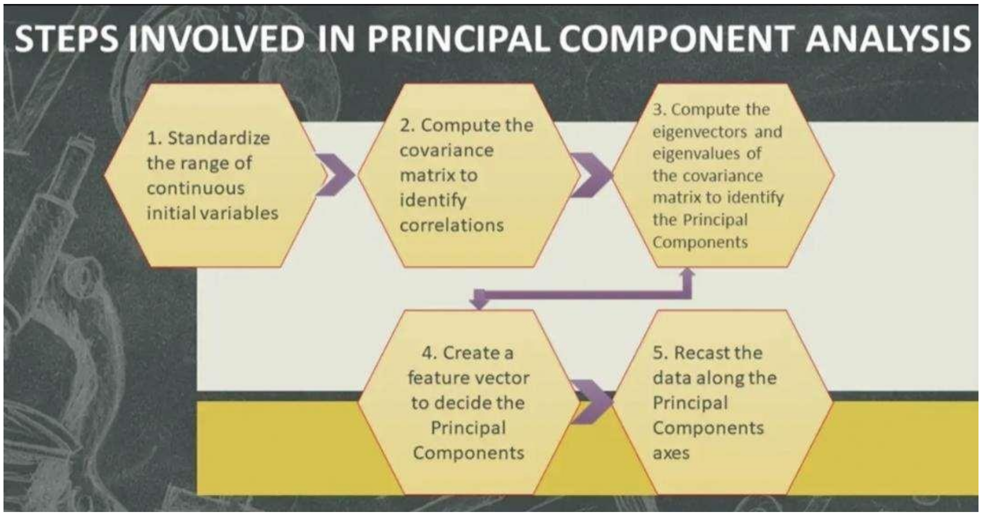


Fig: 3.2.1.2. Steps involved in principal component analysis

3.2.2. METHODOLOGY

The dataset used in this project is taken from Kaggle.com. The dataset contains 148541 number of rows.

The implementation of this project is divided into following steps –

Training the PCA (mode == "train")

3.2.2.1. Data Collection: The first step is to gather a dataset of restaurants that are labeled with corresponding id, name, cost, cuisine, rating etc.

3.2.2.2. Data Preparation: Set up directories for the training and validation datasets, which contain restaurants. Specify the number of training samples (num_train) andvalidation samples (num_val).

3.2.2.3. Model Architecture: Create a PCA model using scikit-learn.

3.2.2.4. Model Compilation: Compile the model with a categorical cross-entropy loss function, the Adam optimizer with a specified learning rate, and accuracy as a metric.

3.2.2.5. Model Training: Train the model using model.fit_generator, Utilize the training and validation data generators and Specify the number of training epochs (num_epoch).

3.2.2.6. Plotting Model History: Call the plot_model_history function to plot the training and validation accuracy and loss curves and Save the generated plots as "plot.png."

Flask Web Application

3.2.2.7. Flask Application Setup:

- A Flask web application was created to provide a user interface for interacting with the restaurant data.

3.2.2.8. Routes and Views:

- Several routes were defined to handle different parts of the application, including the welcome page, homepage, data exploration, data processing, map view, and contact form.

3.2.2.9. User Interface:

- HTML templates were used to render user-friendly web pages, allowing users to explore and interact with the data.

3.2.2.10. Data Filtering:

- Users can specify filter criteria (minimum rating, selected city, and maximum cost) on the 'explore' page, and the application processes these criteria to filter the restaurant data.

3.2.2.11. Principal Component Analysis (PCA):

- PCA was used to reduce the dimensionality of the data and create 2D representations for visualization.

3.2.2.12. Email Integration:

- Flask-Mail was employed to enable users to contact the site administrators through a contact form. Emails are sent to a specified Gmail address.

Visualization

3.2.2.13. Matplotlib and Seaborn:

- Matplotlib and Seaborn were used for creating visualizations, including PCA plots and other charts as needed.

Deployment

3.2.2.14. Running the Application:

- The application is configured to **run in debug mode for development purposes.**

3.2.3. MODULES

3.2.3.1. Preprocessing Description :

Real world data usually have the following drawbacks: Incompleteness, Noisy and Inconsistence. So, these data need to be preprocessed to get the data suitable for analysis purposes, and the preprocessing includes the following tasks:

- **Data cleaning:** fill in missing values, smooth noisy data, identify or remove outliers, andresolve inconsistencies. Data integration: using multiple databases, data cubes, or files.
- **Data transformation:** normalization and aggregation.
- **Data reduction:** reducing the volume but producing the same or similar analytical results.
- **Data discretization:** part of data reduction, replacing numerical attributes with nominal ones.

3.2.3.1 Prediction Description:

For the "Facial Emotional Detection using Deep Learning and AI", the prediction phase involves utilizing the trained Mini-Xception CNN model to analyze real-time video streams from a webcam. The following steps describe how the project makes predictions in real-time:

- **Real-Time Video Analysis:** The project captures live video frames from the webcam, and each frame is processed for facial emotion analysis.
- **Face Detection:** The OpenCV library is used to perform face detection. The Haar Cascade Classifier is applied to locate faces within the video frames.
- **Preprocessing:** Once faces are detected, the grayscale images of faces are resized to 48x48 pixels to match the input size required by the Mini-Xception model. These images are then normalized by rescaling pixel values to the range [0, 1].
- **Emotion Prediction:** The Mini-Xception CNN model is employed to predict the emotional state of the individuals in each frame. It classifies emotions into predefined categories, including "Angry," "Disgusted," "Fearful," "Happy," "Neutral," "Sad," and "Surprised."
- **Display and Database:** The emotions predicted for each frame are displayed on the video feed by adding text labels to the video frames. Simultaneously, the predicted emotion and the processed facial image are saved to a database for further analysis or record-keeping.
- **User Interaction:** The project offers a real-time user interface that can visually represent the analyzed emotions. Users can observe the emotional analysis as graphical representations or textual labels directly on the screen.

3.3. SYSTEM ANALYSIS METHODS

3.3.1. USE CASE DAIGRAM

Use case diagram represent the overall scenario of the system. A scenario is nothing but a sequence of steps describing an interaction between a user and a system. Thus use case is a set of scenario tied together by some goal. The use case diagram are drawn for exposing the functionalities of the system.

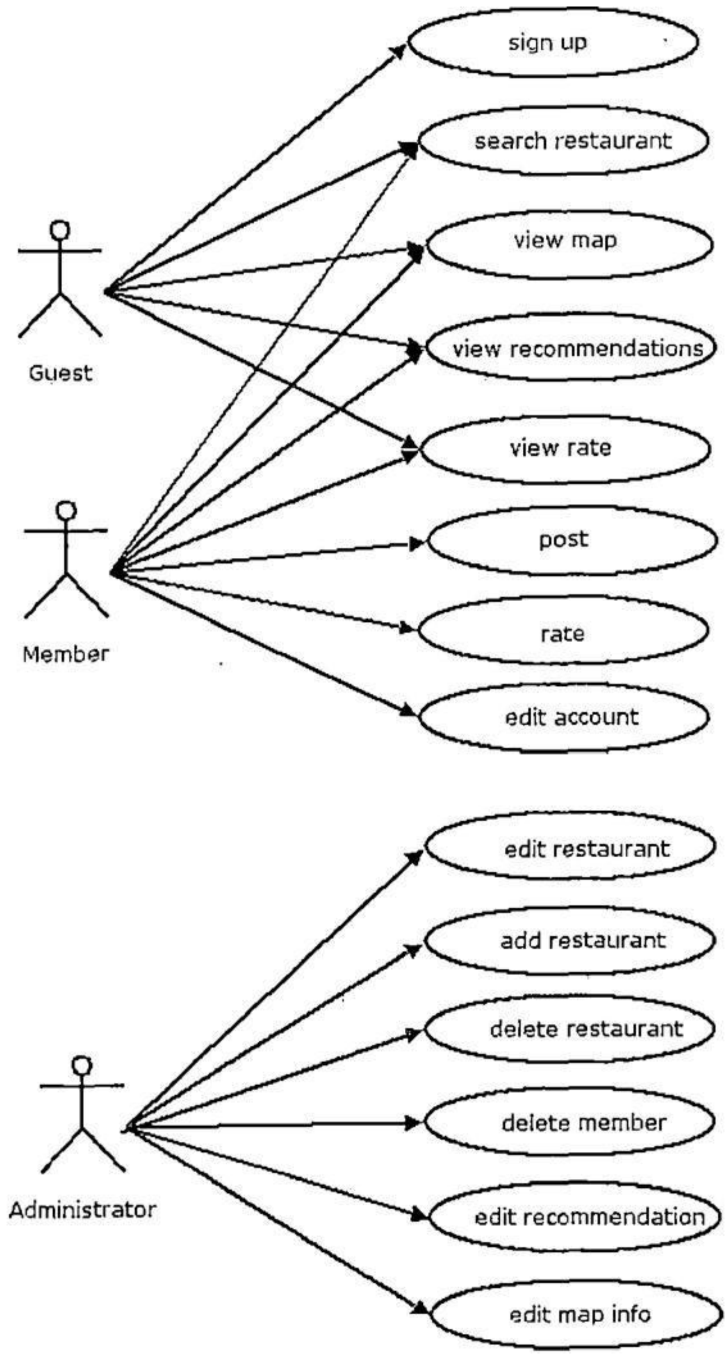


Fig: 3.3.1. USE CASE DIAGRAM

3.3.2. DATAFLOW DIAGRAM

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of business.

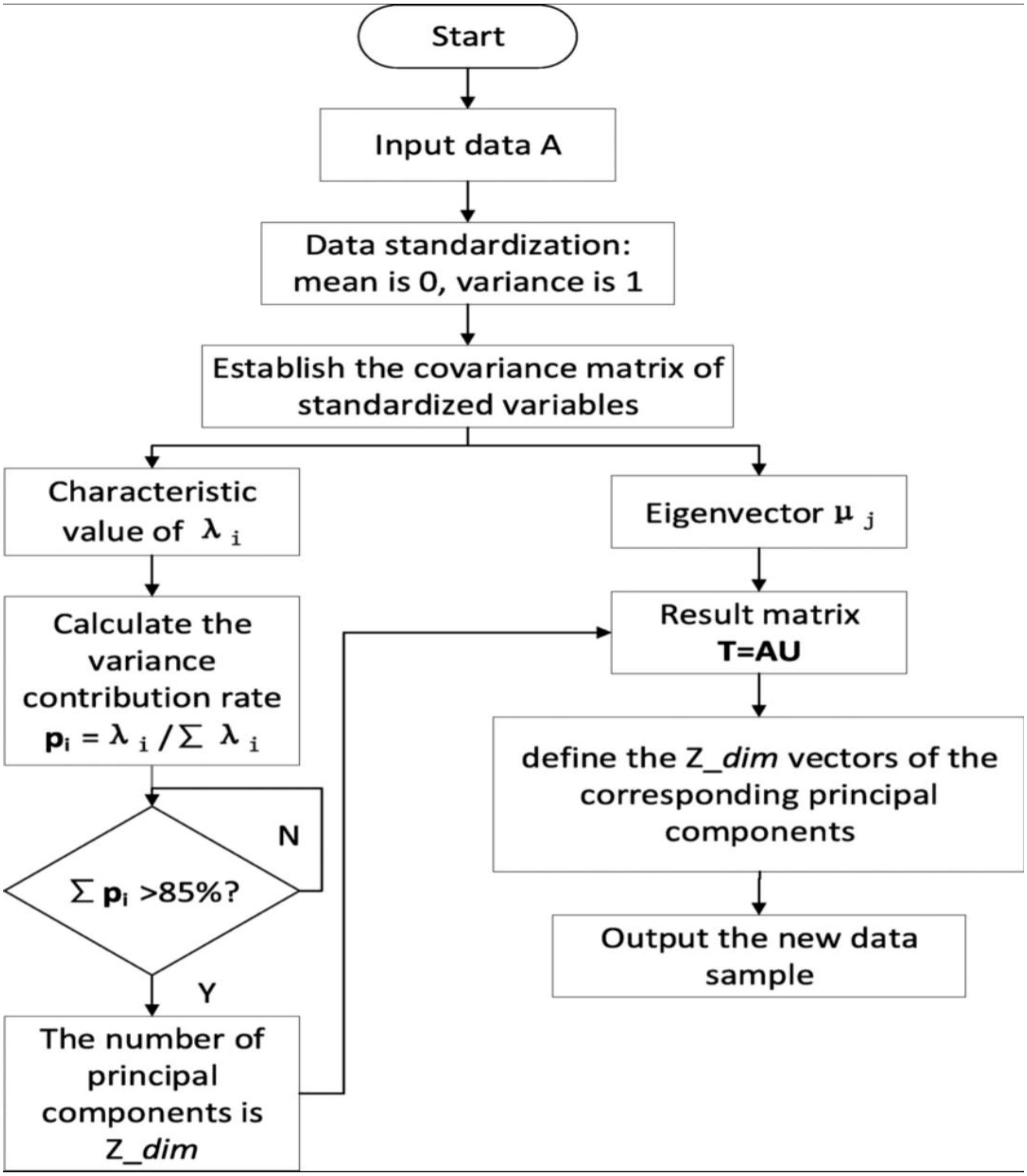


Fig:3.3.2. DATAFLOW DIAGRAM

3.4. SYSTEM DESIGN

3.4.1. SYSTEM ARCHITETURE

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed, design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affection the quality of the software; it has a major impact on the later phase, particularly testing, maintenance. The output of this phase is the design document. This document is similar to a blueprint for the solution and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phases System Design and Detailed Design.

System Design also called top-level design aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided.

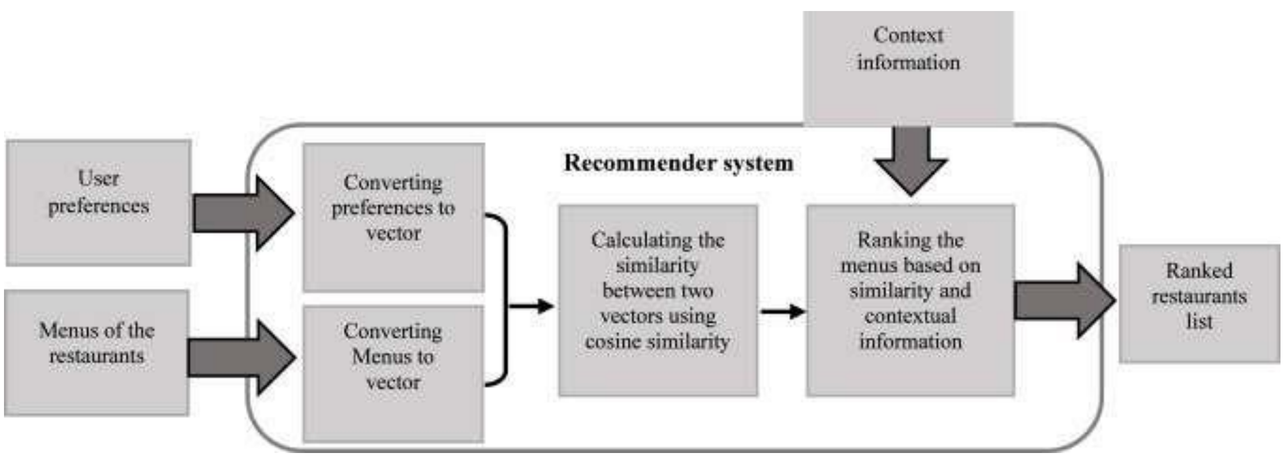


Fig:3.4.1 Food Recommendation System Architecture

3.4.2. CLASS DIAGRAM

Class Diagram gives the static view of an application. A class diagram describes the types of objects in the system and the different types of relationships that exist among them. This modeling method can run with almost all Object- Oriented Methods. A class can refer to another class. A class can have its objects or may inherit from other classes.

- Class Diagram Illustrates data models for even very complex information systems
- It provides an overview of how the application is structured before studying the actual code. This can easily reduce the maintenance time
- It helps for better understanding of general schematics of an application.
- Allows drawing detailed charts which highlights code required to be programmed
- Helpful for developers and other stakeholders.

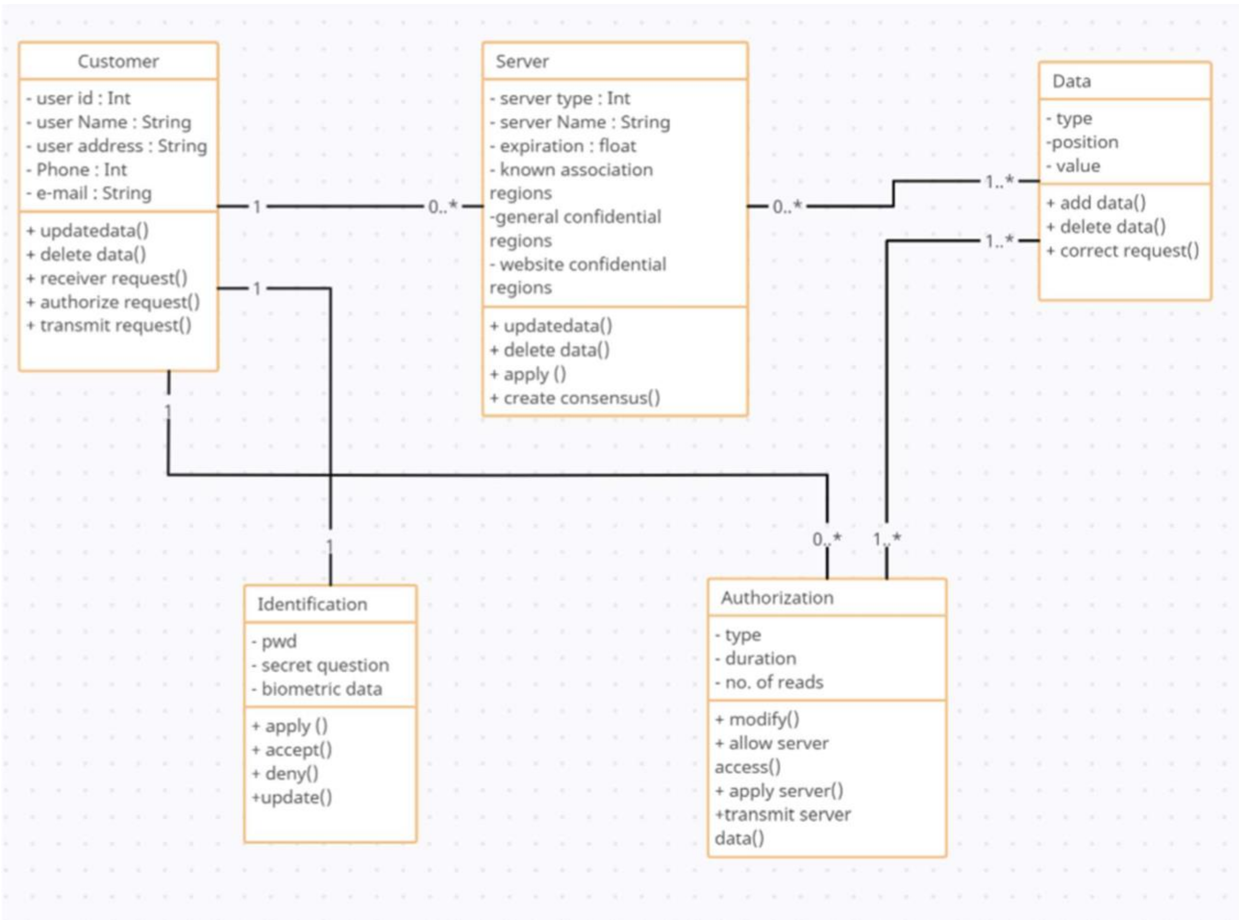


Fig:3.4.2. CLASS DIAGRAM

3.4.3. SEQUENCE DIAGRAM

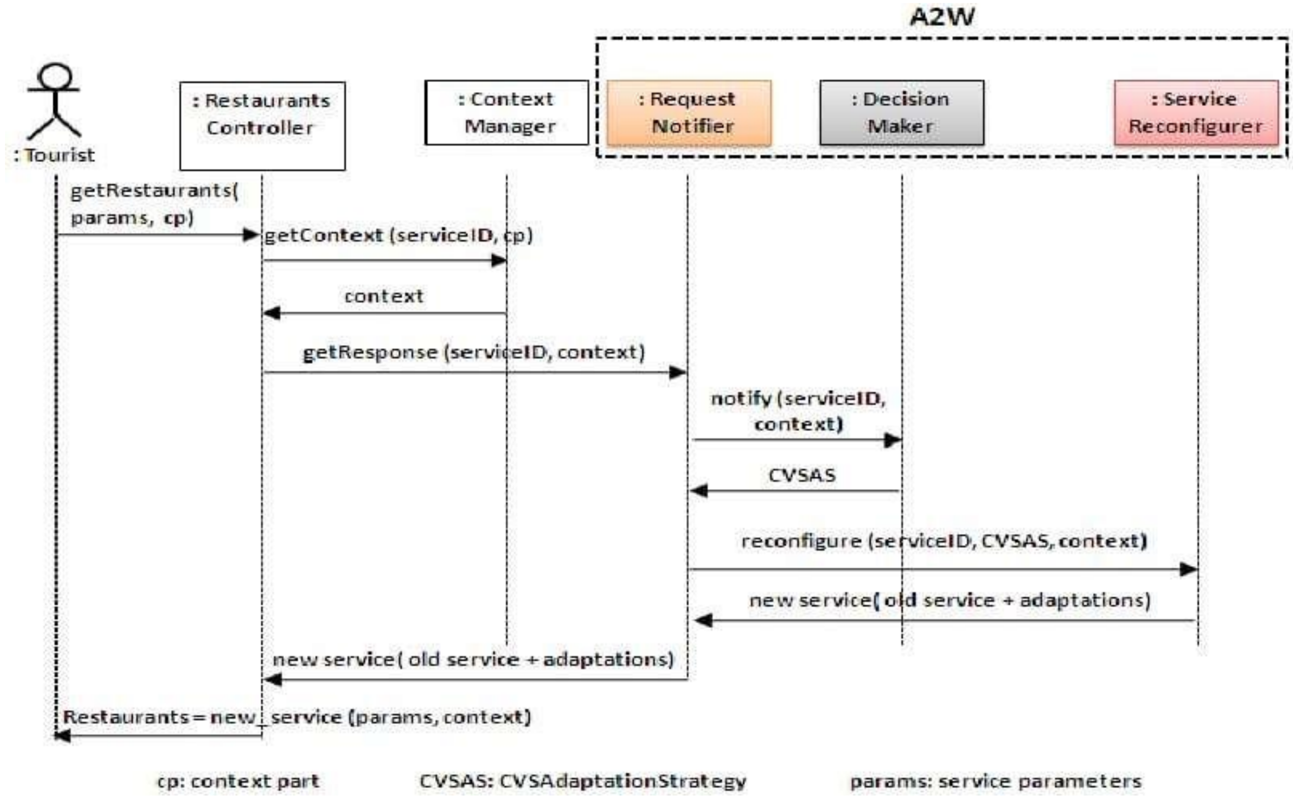


Fig: 3.4.3. SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. Sequence diagrams can be useful references for businesses and other organizations.

Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.

CHAPTER – 4

IMPLEMENTATION

4. IMPLEMENTATION

4.1. TOOLS USED

INTRODUCTION TO PYTHON : Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python Flask:

Flask is a micro web framework for Python that is used to build web applications. It is called a "micro" framework because it provides the essentials for building web applications without imposing too many pre-defined tools or libraries on the developer. This allows developers to have more flexibility and freedom in choosing the components they want to use.

Here are some key features of Flask:

1. Routing: Flask provides a simple and intuitive way to define URL routes and associate them with specific functions, which are called when a specific URL is accessed.
2. Templating: Flask includes a built-in template engine called Jinja2, which allows you to generate dynamic HTML content by combining templates with data.
3. HTTP Methods: You can easily handle different HTTP methods (GET, POST, etc.) for the same URL by defining specific functions to process each method.
4. Request and Response Handling: Flask simplifies handling incoming requests and generating responses. You can access request data, cookies, and form data, and easily return HTML, JSON, or other content as responses.
5. Extensibility: Flask is highly extensible, and you can add various extensions and libraries to enhance its functionality. There's a rich ecosystem of Flask extensions available for tasks like authentication, database integration, and more.
6. Lightweight: Flask is minimalistic and doesn't include a lot of features out of the box, which can be an advantage for developers who prefer to add only what they need.
7. RESTful API Support: Flask is often used to build RESTful APIs due to its simplicity and ease of handling HTTP methods.
8. Good Documentation: Flask has excellent documentation that makes it easy for developers to get started and learn how to use the framework effectively.

Overall, Flask is a great choice for building small to medium-sized web applications, APIs, and prototypes. It's known for its simplicity, flexibility, and the ease with which developers can get started with web development in Python.

History of Python

1. Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands
2. Python is derived from many other languages, including ABC, Modula-3, C, C++,Algol68, SmallTalk, Unix shell, and other scripting languages.
3. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).
4. Python is now maintained by a core development team the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python's standard library: Pandas, Numpy, Sklearn ,Matplotlib ,Tensorflow,cv2,PLI
Importing Datasets

Seaborn:

Seaborn is a popular data visualization library in Python that builds on top of Matplotlib to create attractive and informative statistical graphics. It provides a high-level interface for creating aesthetically pleasing and informative visualizations, making it a valuable tool for data analysis and exploration. Seaborn is known for its ability to generate various types of plots, including scatter plots, bar plots, heatmaps, and violin plots, with just a few lines of code. It also comes with built-in themes and color palettes to enhance the aesthetics of the visualizations. Moreover, Seaborn is particularly well-suited for visualizing relationships between variables and exploring patterns within datasets, making it a popular choice among data scientists and analysts for data visualization tasks.

Pandas:

Pandas is a popular open-source data manipulation and analysis library for Python. It provides data structures and functions for working with structured data, primarily in the form of two main data structures:

1. DataFrame: A 2-dimensional, labeled data structure similar to a spreadsheet or SQL table. It allows you to store and manipulate data in rows and columns, making it ideal for working with structured datasets.

2. Series: A 1-dimensional labeled array capable of holding data of any type. Series are often used to represent a single column or row of data within a DataFrame.

Pandas offers a wide range of data manipulation and analysis tools, including:

- Data cleaning and preparation: Methods for handling missing data, reshaping data, and filtering rows or columns.
- Data exploration: Tools for summarizing and visualizing data, making it easier to understand and gain insights from the data.
- Data merging and joining: Capabilities to combine multiple DataFrames based on common columns or indices.
- Grouping and aggregation: Functions for grouping data and computing summary statistics.

- Time series analysis: Support for time-based data, including date and time manipulation.
- I/O operations: Reading and writing data from various file formats, such as CSV, Excel, SQL databases, and more.

Pandas is a fundamental tool in the field of data analysis and is often used in conjunction with other libraries like NumPy, Matplotlib, and scikit-learn to perform data analysis, data visualization, and machine learning tasks. It simplifies many data-related operations and is widely adopted by data scientists, analysts, and researchers.

Numpy:

Numpy is one such powerful library for array processing along with a large collection of high-level mathematical functions to operate on these arrays. These functions fall into categories like Linear Algebra, Trigonometry, Statistics, Matrix manipulation, etc. Getting NumPy NumPy's main object isa homogeneous multidimensional array. Unlike python's array class which only handles one-dimensional array, NumPy's nd array class can handle multidimensional array and provides more functionality. NumPy's dimensions are known as axes. For example, the array below has 2 dimensions or 2 axes namely rows and columns. Sometimes dimension is also known Page 58 as a rank of that particular array or matrix.

Gmail SMTP Server:

The Gmail SMTP server, or Simple Mail Transfer Protocol server, is a service provided by Google that allows you to send emails from your Gmail account through third-party email clients or applications. SMTP is a protocol used for sending outgoing emails, and Gmail's SMTP server settings enable you to configure your email client to use Gmail as the outgoing mail server. To use Gmail's SMTP server, you need to set it up with your Gmail credentials and a secure connection. Google typically requires authentication, using your Gmail username and password, and SSL/TLS encryption to ensure the security of your email transmissions. Configuring your email client with Gmail's SMTP server details allows you to send messages using your Gmail account while benefiting from Gmail's reliability and deliverability. It's a convenient way to manage your email communication through different applications while keeping everything synchronized with your Gmail account.

Matplotlib:

Matplotlib is a widely used Python library for creating high-quality, customizable visualizations and plots. It provides a versatile platform for generating a wide range of charts, graphs, and figures, making it an essential tool for data visualization and scientific plotting. Matplotlib offers straightforward interface, allowing users to create various types of plots, from simple line charts to complex 3D visualizations. With its extensive customization options, you can tailor every aspect of your plots, such as colors, labels, and annotations. Whether you're analyzing data, exploring trends, or presenting results, Matplotlib empowers you to create informative and visually appealing graphics that enhance your understanding and communication of data.

FPDF:

FPDF (Free PDF) is a popular open-source PHP library that allows developers to PDF with text, images, and basic graphics. FPDF is widely used in web development to generate invoices, reports, and other printable documents. The library offers various features like support for different fonts, page formatting, and basic drawing functions. Developers can use FPDF to generate PDFs by defining the document's structure and content programmatically. Its ease of use and robust functionality make it a valuable tool for generating PDFs In PHP applications.

Scikit-Learn:

Scikit-learn, often abbreviated as sklearn, is a popular open-source machine learning library for Python. It provides a wide range of tools and functionalities for various machine learning tasks, such as classification, regression, clustering, dimensionality reduction, and more. Scikit-learn is built on top of other Python libraries like NumPy, SciPy, and Matplotlib, making it easy to integrate into the Python data science ecosystem. It offers a consistent and user-friendly API for a diverse set of machine learning algorithms and data preprocessing techniques. Scikit-learn is widely used by data scientists and researchers for its simplicity and efficiency in developing and deploying machine learning models, making it an essential tool for anyone working in the field of data analysis and predictive modeling.

WTForms:

WTForms is a powerful and popular Python library used for handling web forms in web applications, particularly in conjunction with web frameworks like Flask and Django. It simplifies the process of form creation, validation, and rendering in web applications. With WTForms, you can define form fields and their validation rules as . Python classes, making it easy to generate HTML forms, validate user input, and render error messages. This approach promotes code reusability, maintainability, an security by ensuring that user input conforms to the desired format and constraints. WTForms supports a wide range of field types, including text fields, checkboxes, file uploads, and more, and allows you to create complex and customizable forms efficiently. It's a valuable tool for enhancing the user experience and ensuring data integrity.

Rooting :

Routing in Flask is the process of mapping URLs to specific functions that handle requests for those URLs. This allows you to organize your application's code and make it easier to maintain. To create a route in Flask, you use the `app.route()` decorator. The first argument to the `app.route()` decorator is the URL path, and the second argument is the function that will handle requests for that URL.

You can also use regular expressions to define routes. This is useful for matching URLs that have dynamic parts.You can also import view functions from other modules and use them with routes. This is useful for organizing your code.Routing is an essential part of any Flask application. It allows you to organize your application's code and make it easier to maintain.

Error Handling in Flask:

Error handling is an essential aspect of web development, and Flask provides robust mechanisms for managing errors that arise during application execution. It ensures a seamless user experience and prevents application crashes.

- Utilize appropriate error handling methods to ensure a pleasant user experience.
- Provide informative error messages that clearly convey the issue to the user.
- Implement logging mechanisms to track and analyze errors for improvement purposes.
- Employ proactive error prevention strategies to minimize error occurrences.

4.2. PSEUDO CODE

Load the data:

```
# Load and preprocess data
default_dataset_path = 'D:\\Downlodes\\sam\\dataset.csv'
raw_data = pd.read_csv(default_dataset_path)
raw_data_frame = pd.DataFrame(raw_data)
raw_data_frame['rating'] = pd.to_numeric(raw_data_frame['rating'], errors='coerce')
raw_data_frame.dropna(inplace=True)
label_encoder = LabelEncoder()
food_rating_encoded = label_encoder.fit_transform(raw_data_frame['rating'])
numeric_columns = ['cost', 'id', 'rating']
numeric_data = raw_data_frame[numeric_columns]

for column in numeric_columns:
    numeric_data[column] = pd.to_numeric(numeric_data[column], errors='coerce')

numeric_data.dropna(inplace=True)
data_scaler = StandardScaler()
scaled_data_frame = data_scaler.fit_transform(numeric_data)
```

Python File:

```
from flask import Flask, render_template, request,jsonify,Response
import pandas as pd
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas
from io import BytesIO
from flask import jsonify
import warnings
import base64
import matplotlib.pyplot as plt
import seaborn as sns
from fpdf import FPDF
```

```
app = Flask(_name_)
warnings.filterwarnings("ignore",  category=Warning)

# Sample data for dropdowns
rating_values =  sorted(raw_data_frame['rating'].unique(), reverse=True)
city_values = sorted(raw_data_frame['city'].unique(), reverse=False)
cost_values = sorted(raw_data_frame['cost'].unique(), reverse=False)

@app.route('/')
def welcome():
    return  render_template('welcome.html')

@app.route('/home')
def home():
    return  render_template('home.html')

@app.route('/explore')
def explore():
    return render_template('explore.html', rating_values=rating_values, city_values=city_values,
cost_values=cost_values)

@app.route('/process_data', methods=['POST'])
def process_data():
    try:
        min_rating  = float(request.form.get('min_rating'))
        selected_city = request.form.get('selected_city')
        max_cost = float(request.form.get('max_cost'))
        selected_classifier  =  request.form.get('selected_classifier')

        filtered_data  =  raw_data_frame[(raw_data_frame['rating']  >=  min_rating)  &
                               (raw_data_frame['city'] == selected_city) &
                               (raw_data_frame['cost'] <= max_cost)]

        if not filtered_data.empty:
            # Apply PCA
            pca = PCA(n_components=2)
            # You can change the number of components as needed
            pca_data = pca.fit_transform(scaled_data_frame)

            # Transform filtered data using PCA
            filtered_pca_data  =  pca.transform(filtered_data[numeric_columns])
```

```
        filtered_pca_results = []
        for i, row in enumerate(filtered_data.iterrows()):
            filtered_pca_results.append({
                "name": row[1]['name'],
                "rating": row[1]['rating'],
                "city": row[1]['city'],
                "cost": row[1]['cost'],
                "cuisine": row[1]['cuisine'],
                "address": row[1]['address'],
                "link": row[1]['link'],
            })

        return  render_template('filtered_results.html',  filtered_results=filtered_pca_results)

      else:
        return render_template('filtered_results.html', filtered_results=[], selected_classifier=selected_classifier)
  except Exception as e:
      return jsonify({"error": str(e)})

@app.route('/map_view', methods=['GET','POST'])
def map_view():
    return  render_template('map_view.html')

from flask import Flask, render_template, request, flash, redirect, url_for
from flask_mail import Mail, Message
from flask_wtf import FlaskForm
from wtforms import StringField, TextAreaField, SubmitField, validators
from flask_wtf.csrf import CSRFProtect

# Initialize CSRF protection and disable it for the entire app
csrf = CSRFProtect(app)
csrf.init_app(app)
app.config['WTF_CSRF_ENABLED'] = False
# Disable Flask sessions
app.config['SESSION_TYPE'] = 'filesystem'
app.config['SESSION_PERMANENT']  = False

# Configure Flask-Mail for Gmail
app.config['MAIL_SERVER'] = 'smtp.gmail.com'
app.config['MAIL_PORT'] = 587
app.config['MAIL_USE_TLS'] = True
```

```
app.config['MAIL_USE_SSL'] = False
app.config['MAIL_USERNAME'] = 'sdmunwarali786@gmail.com'
app.config['MAIL_PASSWORD'] = 'Munwarali@2002'
mail = Mail(app)

# Define the ContactForm class
class  ContactForm(FlaskForm):
    name = StringField('Name', [validators.DataRequired()])
    email = StringField('Email', [validators.DataRequired(), validators.Email()])
    message = TextAreaField('Message', [validators.DataRequired()])
    submit = SubmitField('Send')

# Contact route with CSRF protection and sessions disabled
@app.route('/contact', methods=['GET', 'POST'])
def contact():
    form = ContactForm()
    if form.validate_on_submit():
        name = form.name.data
        email = form.email.data
        message = form.message.data

        msg = Message('Contact Us Form Submission', sender='sdmunwarali786@gmail.com',
recipients=['sdmunwarali786@gmail.com'])
        msg.body = f'Name: {name}\nEmail: {email}\nMessage: {message}'

        try:
            mail.send(msg)
            flash('Message sent successfully!', 'success')
            return redirect(url_for('contact'))
        except Exception as e:
            flash(f'Error: {str(e)}', 'danger')

    return  render_template('contact.html',  form=form)

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

User interfaces :

```
<!DOCTYPE html>
<html lang="en">
<head class="head">
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>FOOD RECOMENDATION System</title>
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css"></link>

   <script>
      function showIt(item){
      var items = document.getElementsByClassName('items')
      item = document.getElementById(item)
      for(let i=0; i< items.length; i++)
      {
         items[i].style.display = 'none'
      }
      item.style.display = 'block'
   }
   </script>
</head>

<body >
   <div class="grid-container">
      <div class="nav">
   <nav>
      <ul>
         <li class="active" >
            <a onclick="showIt('home')" href="#" >
               <i class="fa fa-home" aria-hidden></i>
               <span class="title">Home</span>
            </a>
         </li>
         <li >
            <a onclick="showIt('explore')" href="#" >
               <i class="fa fa-filter" aria-hidden></i>
               <span class="title">Explore</span>
            </a>
         </li>
         <li >
            <a onclick="showIt('map')" href="#">
               <i class="fa fa-map-signs" aria-hidden></i>
               <span class="title">Map</span>
            </a>
         </li>
         <li >
            <a onclick="showIt('contact')" href="#">
               <i class="fa fa-comment" aria-hidden></i>
               <span class="title" style="font-size: 15px;font-weight: bold;">Contact us</span>
            </a>
         </li>
         <li >
```

```
            <a onclick="showIt('about')" href="#">
                <i class="fa fa-info-circle" aria-hidden></i>
                <span class="title">About</span>
            </a>
        </li>
        <div class="selected-option-bg"></div>
      </ul>
    </nav>
    </div>
    <br><br><br>

    <div class="content">
      <div class="sec">
        <iframe class="items" src="{{url_for('explore')}}" id="explore" style="width: 100%; height:
100%; border: none;"></iframe>
        <iframe class="items"
src="https://maps.google.com/maps?q=india&t=&z=10&ie=UTF8&iwloc=&output=embed" id="map"
style="width: 100%; height: 100%; border: none;"></iframe>
        <iframe class="items" src="{{url_for('contact')}}" id="contact" style="width: 100%; height:
100%; border: none;"></iframe>

      </div>
    </div>
</div>

</body>

<script>
    const list = document.querySelectorAll('li');


function activeLink() {
    list.forEach((item) => item.classList.remove('active'));
    this.classList.add('active')
}

list.forEach((item) => item.addEventListener('click', activeLink));
</script>
</html>
<style>

    *{
    margin: 1;
    padding: 0;
    box-sizing: border-box;
    font-family: sans-serif;
    align-items: top;
}
    .grid-container {
    display: grid;
    grid-template-rows: auto 1fr;
    min-height: 110vh;
    background: var(--bgc);
    width: 99%;
    margin-top: 10px;
    max-width: 1000vw;
}
```

```
.content {
   background-color:white;
   padding: 0px;
   border-radius:  15px;
   box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
   max-width: 1000vw;
   margin-bottom: 10px;
}

.form-wrapper {

  display: flex;
  justify-content: center;
  align-items: center;
  width: 99%;
  margin-top: 10px;


}
.sec {

  padding: 0px;
  border-radius: 15px;
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
  width: 100%;
  max-width: 1000pw;
  margin: 0 auto;
  height: 100vh;
  overflow: auto;
  background-color:white;
  text-align: center;
}


.items{
   display: none;
}

#home{
   display: block;
}


:root{
   --bgc:#1d1a39;
}

body{
   margin-top: 50px;

   display: flex;
   justify-content: center;
   align-items:top;
   height: 100vb;
   background: var(--bgc);
}
```

```
nav{

    display: flex;
    justify-content: center;
    align-items: center;
    background: white;
    border-radius: 15px;
    position: relative;
    width: 500px;
    height: 80px;
    margin-bottom: -30px;
}

.nav{
    display: flex;
    justify-content: center;
    width: 100%;

}

nav ul{
    display: flex;
    justify-content: space-between;
    width: 400px;
}

nav ul li{
    position: relative;
    list-style: none;
    width: 90px;
    z-index: 1;
}

ul li a{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    text-align: center;
    text-decoration: none;
    position: relative;
    width: 100%;
}

.fa{
    position: relative;
    display: block;
    line-height: 80px;
    font-size: 30px;
    transition: 0.4s;
    color: black;
    text-align: center;
}

ul li.active a .fa{
    transform: translateY(-40px);
    color:white;
```

```
}

ul li.active a .title{
    opacity: 1;
    transform: translateY(13px);
}

.title{
    position: absolute;
    font-weight: bold;
    font-size: 18px;
    color: black;
    opacity: 0;
    transition: 0.4s;
    transform: translateY(10px);
}

.selected-option-bg{
    position: absolute;
    width: 80px;
    height: 80px;
    top: -50%;
    border-radius: 50%;
    background: linear-gradient(30deg, #e82ac5, #7d2ae8);
    transition: 0.4s;
    border: 5px solid var(--bgc);
}

.selected-option-bg::before{
    content: "";
    position: absolute;
    top: 50%;
    left: -18px;
    width: 15px;
    height: 19px;
    background-color: white;
    border-top-right-radius: 20px;
    box-shadow: 0px -8px 0 0 var(--bgc);
}

.selected-option-bg::after{
    content: "";
    position: absolute;
    top: 50%;
    right: -18px;
    width: 15px;
    height: 19px;
    background-color: white;
    border-top-left-radius: 20px;
    box-shadow: 0px -8px 0 0 var(--bgc);
}

ul li:nth-child(1).active ~ .selected-option-bg{
    transform: translateX(calc(80px * 0));
}

ul li:nth-child(2).active ~ .selected-option-bg{
```

```
    transform: translateX(calc(80px * 1));
}

ul li:nth-child(3).active ~ .selected-option-bg{
    transform: translateX(calc(80px * 2));
}

ul li:nth-child(4).active ~ .selected-option-bg{
    transform: translateX(calc(80px * 3));
}

ul li:nth-child(5).active ~ .selected-option-bg{
    transform: translateX(calc(80px * 4));
}
</style>
```

Results Generatation code:

```
<!DOCTYPE html>
<html>
<head>
    <title>Filtered Results</title>
    <style>
            .fade-in {
            opacity: 0;
            animation: fadeInAnimation 1s forwards;
        }

        @keyframes fadeInAnimation {

            from{
                from {
                opacity: 0;
                transform: translateY(50px);
            }
            }
            to {
                opacity: 1;
            }
        }

        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #eaeaea;
        }
        h1 {
            text-align: center;
            margin-top: 20px;
        }
```

```
table {
    border-collapse: collapse;
    box-shadow: 10px 10px 20px rgba(0, 0, 0, 0.1);
    border-radius: 10px;
    overflow: hidden; /* Hide border-radius overflow */
    animation: fadeInUp 1s ease;
    background-color: #ffffff;
    height: 100%;

}

.header-container {
    text-align: center;
    margin-bottom: 20px;
}
.table-container {
    display: flex;
    justify-content: center;
    align-items: center;


}
/* Use a lighter background color for the PDF version */
table.pdf-version {
    font-size: 16px; /* Adjust font size for better visibility in PDF */
    border-collapse: collapse;
    width: 98%;
    background-color: #fff; /* Use a lighter background color */
}
/* Apply similar styles for table cells */
table.pdf-version th,
table.pdf-version td {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: center;
    background-color:white; /* Use a slightly darker background for cells */
    color: #333; /* Adjust text color for better contrast */
}
table.pdf-version th {
    background-color: #f39f5a;
    color: white;
}

button{
background-color: aliceblue;
border-color: #384358;
border-style: solid;
border-width: 1px;
color:#384358;
```

```
    height: 20px;
    width: 50x;
    border-radius: 3px;
    font-size: 15px;
    cursor:pointer;
    margin-right: 30px;
    transition: background-color 0.7s;
    padding-left: 15px;
    padding-right: 15px;
    padding-top: 5px;
    padding-bottom: 24px;
}
button:hover{
    color: aliceblue;
    background-color: #06193d;

}
.pad{
    padding-bottom: 20px;
}

    </style>
    <script  src="https://cdnjs.cloudflare.com/ajax/libs/html2pdf.js/0.10.1/html2pdf.bundle.min.js"></script>

</head>
<body >
    <h1 style="color: red;font-family:Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;font-
stretch:wider;padding-right: 30px ;"><b>Filtered Results</b></h1>


<div class="pad">
    <div class="table-container" class="download_but"><center>
        <button onclick="downloadAsPDF()">Download Results as PDF</button>
    </center>
    </div></div>

    <div class="table-container fade-in">
        {% if filtered_results %}
            <table class="pdf-version" border="1">
                <thead>
                    <tr>
                        <th>Restaurant Name</th>
                        <th>Rating</th>
                        <th>City</th>
                        <th>Cost</th>
                        <th>Cuisine</th>
                        <th>Address</th>
                        <th>Links to order your Food</th>
                    </tr>
```

```
            </thead>
            <tbody>
                {% for result in filtered_results %}
                    <tr>
                        <td>{{ result['name'] }}</td>
                        <td>{{ result['rating'] }}</td>
                        <td>{{ result['city'] }}</td>
                        <td>{{ result['cost'] }}</td>
                        <td>{{ result['cuisine'] }}</td>
                        <td>{{ result['address'] }}</td>
                        <td><a href="{{ result['link']}}" target="_blank" style="text-decoration:
none;color:rgb(21, 1, 79);">{{ result['link']}}</a></td>
                    </tr>
                {% endfor %}
            </tbody>
          </table>
        {% else %}
          <p>No results found for the selected criteria.</p>
        {% endif %}
    </div>

    <script>
        function downloadAsPDF() {
            const table = document.querySelector('.pdf-version');
            if (table) {
                html2pdf().from(table).set({
                    margin: 10,
                    filename: 'filtered_results.pdf',
                    image: { type: 'jpeg', quality: 0.99 },
                    html2canvas: { scale: 1 },
                    jsPDF: { unit: 'mm', format:'a3', orientation:'landscape' }
                }).save();
            } else {
                alert('No table found.');
            }
        }
    </script>
</body>
</html>
```

4.3. PACKAGE DIAGRAM

Package diagram, a kind of structural diagram, shows the arrangement and organization of model elements in middle to large scale project. Package diagram can show both structure and dependencies between sub-systems or modules, showing different views of a system, for example, as multi-layered application – multi-layered application model.

Components of Component Diagram:

Main Components:

User : User gives input and access the output. Restaurant suggestions

Suggestion Framework: This is the core component responsible for the suggested system.

User Interface: Represents the interface for displaying the suggested restaurants, location of restaurants and online ordering.

Restaurant suggestions: Gives the nearest restaurant suggestions based on the given input.

Location: user can access the location option where the suggested restaurants location are present.

Online Ordering: online ordering responsible for ordering the food in selected restaurant.

Database: Represents the database where list of restaurants data is stored.

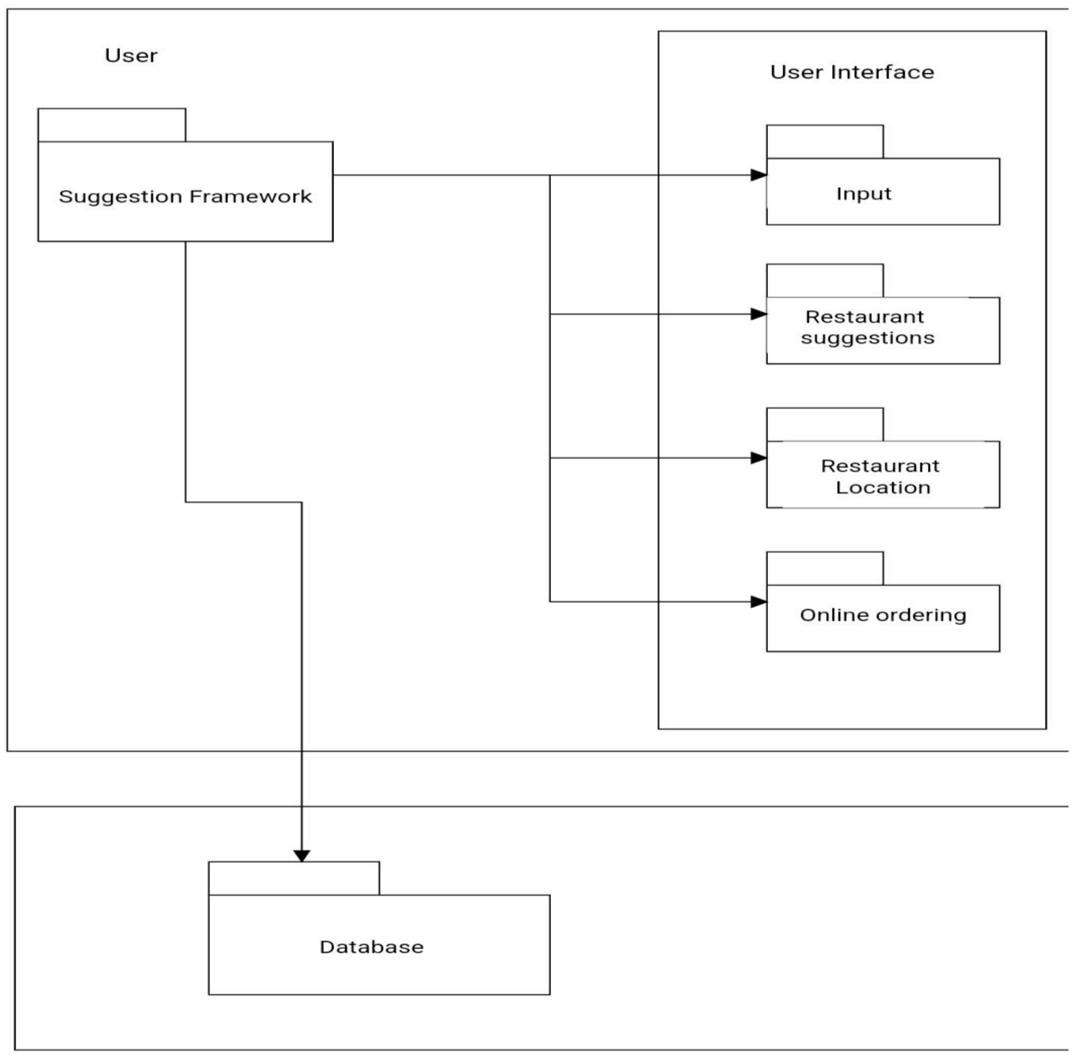


Fig: 4.3. PACKAGE DIAGRAM

4.4. DEPLOYMENT DIAGRAM

In the context of the Unified Modeling Language (UML), a deployment diagram falls under the structural diagramming family because it describes an aspect of the system itself. In this case, the deployment diagram describes the physical deployment of information generated by the software program on hardware components.

- Deployment diagrams have several valuable applications. You can use them to:
- Show which software elements are deployed by which hardware elements.
- Illustrate therun time processing for hardware.
- Provide a view of the hardware system's topology.

Deployment Diagram Elements:

Variety of shapes makes up deployment diagrams. This list offers an overview of the basic elements you may encounter, and you can see most of these items illustrated in the image below

- **Artifact:** A product developed by the software, symbolized by a rectangle with the nameand the word ―artifact‖ enclosed by double arrows.
- **Association:** A line that indicates a message or other type of communication between nodes.
- **Component:** A rectangle with two tabs that indicates a software element.
- **Dependency:** A dashed line that ends in an arrow, which indicates that one node orcomponent, is dependent on another.
- **Interface:** A circle that indicates a contractual relationship. Those objects that realize the interface must complete some sort of obligation.
- **Node:** A hardware or software object, shown by a three-dimensional box.
- **Node as container:** A node that contains another node inside of it—such as in the examplebelow, where the nodes contain components

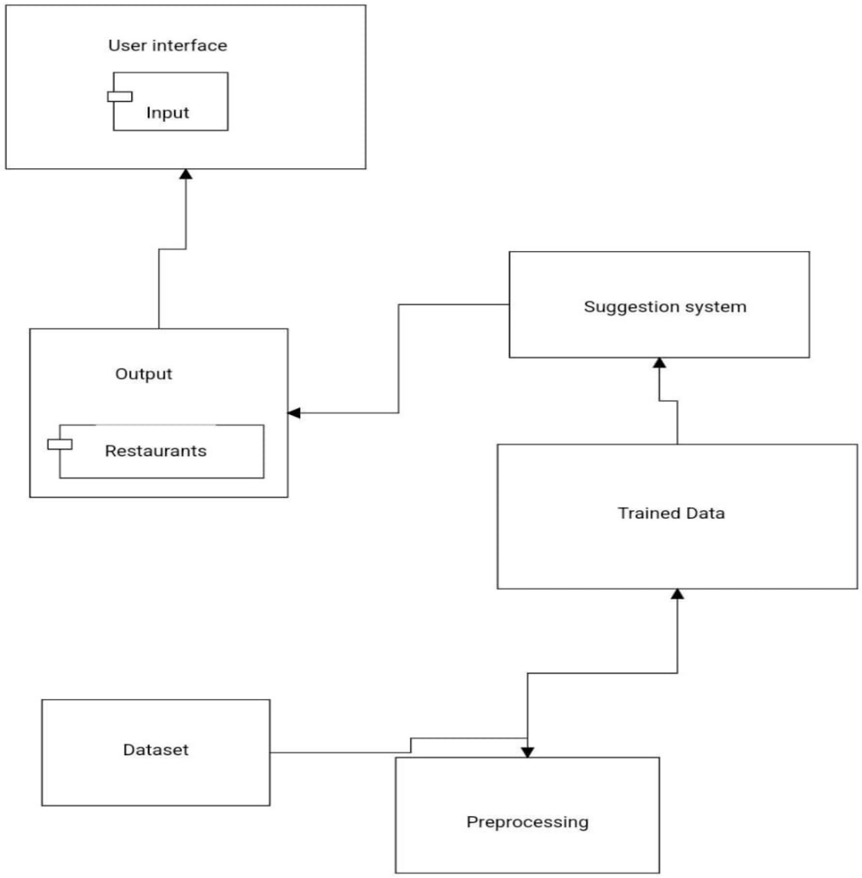


Fig: 4.4. DEPLOYMENT DIAGRAM

.

CHAPTER -5

SCREEN SHOTS

5. SCREEN SHOTS

id	name	city	rating	cost	cuisine	address	link
567335	AB FOODS POINT	Abohar	3.5	200	Beverages,Pizzas	AB FOODS POINT, NEAR RISHI NARANG DENTAL CLINIC , NEAR IDBI BANK, A	https://www.swiggy.com/restaurants/ab-foods-point-central-abohar-abohar-567335
531342	Janta Sweet House	Abohar	4.4	200	Sweets,Bakery	Janta Sweet House, Bazar No.9, Circullar Road, Abohar	https://www.swiggy.com/restaurants/janta-sweet-house-central-abohar-abohar-531342
158203	theka coffee desi	Abohar	3.8	100	Beverages	theka coffee desi, sahtiya sadan road city	https://www.swiggy.com/restaurants/theka-coffee-desi-central-raipura-abohar-158203
187912	Singh Hut	Abohar	3.7	250	Fast Food,Indian	Singh Hut, CIRCULAR ROAD NEAR NEHRU PARK ABOHAR	https://www.swiggy.com/restaurants/singh-hut-nehru-park-main-bazar-abohar-187912
543530	GRILL MASTERS	Abohar	3.5	250	Italian American,Fast Food	GRILL MASTERS, ADA Heights, Abohar - Hanumangarh Rd, Raipura, Abohar,	https://www.swiggy.com/restaurants/grill-masters-central-abohar-abohar-543530
158204	Sam Uncle	Abohar	3.6	200	Continental	Sam Uncle, hanumangarh road near raja bajaj showroom abohar	https://www.swiggy.com/restaurants/sam-uncle-central-main-bazar-abohar-158204
156588	shere punjab veg	Abohar	4	150	North Indian	shere punjab veg, major surinder chowk near verma sons petrol pump and	https://www.swiggy.com/restaurants/shere-punjab-veg-central-krishna-nagri-abohar-156588
244866	Shri Balaji Vaishno Dhaba	Abohar	3.5	100	North Indian	Shri Balaji Vaishno Dhaba, St no 13,6th chowk,main Bazar,near jain bhawar	https://www.swiggy.com/restaurants/shri-balaji-vaishno-dhaba-central-main-bazar-abohar-244866
156602	Hinglaj Kachori Bhandhar	Abohar	4.2	100	Snacks,Chaat	Hinglaj Kachori Bhandhar, street no 11 circular road ,Abohar	https://www.swiggy.com/restaurants/hinglaj-kachori-bhandhar-central-main-bazar-abohar-156602
158193	yummy hub	Abohar	3.5	200	Indian	yummy hub, hanumangarh road near dr naveen sethi hospitalabohar	https://www.swiggy.com/restaurants/yummy-hub-central-abohar-abohar-158193
407249	CHAWLA SAAB THE JUICE MASTER	Abohar	3.5	300	Juices,Beverages	CHAWLA SAAB THE JUICE MASTER, SAHITYA SADAN MARKET, ABOHAR , Al	https://www.swiggy.com/restaurants/chawla-saab-the-juice-master-central-abohar-abohar-407249
156590	Sethi Milk Badam	Abohar	4.2	100	Sweets,Desserts	Sethi Milk Badam, main bazar street no 11 abohar	https://www.swiggy.com/restaurants/sethi-milk-badam-central-krishna-nagri-abohar-156590
338749	Swastik Dhaba	Abohar	3.5	200	North Indian	Swastik Dhaba, Gaushala road street no 22 near shiv mandir abohar	https://www.swiggy.com/restaurants/swastik-dhaba-central-krishna-nagri-abohar-338749
156601	Jodhpuri Kachori	Abohar	3.5	100	Snacks	Jodhpuri Kachori, mandi 1 street no 4 main bazar near sbi bank abohar	https://www.swiggy.com/restaurants/jodhpuri-kachori-central-main-bazar-abohar-156601
156587	Bharawan Da Dhaba	Abohar	4.4	300	Indian	Bharawan Da Dhaba, rimpy bakery 12 circular road main bazar abohar	https://www.swiggy.com/restaurants/bharawan-da-dhaba-central-abohar-abohar-156587
158192	Tandoori Nights	Abohar	3.5	300	Tandoor	Tandoori Nights, near civil hospital canteen abohar	https://www.swiggy.com/restaurants/tandoori-nights-central-abohar-abohar-158192
394554	Roll Express	Abohar	4	200	Fast Food	Roll Express, Roll Express, Opposite Dr. Saria Sethi Hospital, Near DAV Colle	https://www.swiggy.com/restaurants/roll-express-central-abohar-abohar-394554
158195	wah ji waah veg and non veg corner	Abohar	3.5	200	North Indian,Chinese	wah ji waah veg and non veg corner, Hanuman garh road, Down the flyove	https://www.swiggy.com/restaurants/wah-ji-waah-veg-and-non-veg-corner-central-canal-colony-abohar-158195
257181	Shri Balaji fast food and Variety store	Abohar	3.5	100	Indian	Shri Balaji fast food and Variety store, opposite Loona Nursing home	https://www.swiggy.com/restaurants/shri-balaji-fast-food-and-variety-store-central-model-town-abohar-257181
530909	FOODY MOOD	Abohar	4.7	300	Fast Food,Chinese	FOODY MOOD, Arya nagar St no 7 Abohar, Abohar (M Cl), Fazilka 1(Aboha	https://www.swiggy.com/restaurants/foody-mood-central-abohar-abohar-530909
573649	PUNJABI TADKA	Abohar	3.5	700	Punjabi,North Indian	PUNJABI TADKA, HANUMANGARH ROAD,UNDER FLYOVER,OPP SETHI REGE	https://www.swiggy.com/restaurants/punjabi-tadka-central-abohar-abohar-573649
573656	PUNJABI TADKA CHICKEN HUB	Abohar	3.5	650	North Indian,Tandoor	PUNJABI TADKA CHICKEN HUB, HANUMANGARH ROAD,UNDER FLYOVER,O	https://www.swiggy.com/restaurants/punjabi-tadka-chicken-hub-central-abohar-abohar-573656
420675	Royal Chicken	Abohar	4.2	200	Mughlai,North Indian	Royal Chicken, Near Naveen Sethi Hospital, Under Flyover, Abohar	https://www.swiggy.com/restaurants/royal-chicken-central-abohar-abohar-420675
427610	Just Baked	Abohar	3.5	300	Beverages,Pizzas	Just Baked, New Abadi, Ward no. 22, Abohar (M Cl), Fazilka 1(Abohar), Fazi	https://www.swiggy.com/restaurants/just-baked-central-abohar-abohar-427610
435582	Picado International Food	Abohar	3.5	300	Pizzas,Beverages	Picado International Food, Opp New Grain Market, Fazilka Road, Abohar	https://www.swiggy.com/restaurants/picado-international-food-central-abohar-abohar-435582
382204	PubG Cafe	Abohar	3.5	200	Chinese,Pizzas	PubG Cafe, St. No 2, Near Essar petrol pump, Sitto road, Model Town , Abc	https://www.swiggy.com/restaurants/pubg-cafe-central-abohar-abohar-382204
460692	Verma Dhaba	Abohar	3.5	300	Indian	Verma Dhaba, GangaNagar Road, Near Tara Estate, Abohar	https://www.swiggy.com/restaurants/verma-dhaba-central-abohar-abohar-460692
368328	Fresh Food Cafe	Abohar	3.5	150	North Indian	Fresh Food Cafe, Sito Road, Near Railway Crossing, Model Town, Abohar	https://www.swiggy.com/restaurants/fresh-food-cafe-central-model-town-abohar-368328
530878	NAVU PIZZA HOUSE	Abohar	3.5	300	Pizzas,Fast Food	NAVU PIZZA HOUSE, COLLEGE ROAD ABOHAR,ABOHAR,FAZILKA 1,FAZILKA	https://www.swiggy.com/restaurants/navu-pizza-house-central-abohar-abohar-530878
171675	Domino's Pizza	Abohar	4.4	400	Pizzas	Domino's Pizza, ADAHeights,HanumangarhRoad,SunderNagari,Abohar,Dist	https://www.swiggy.com/restaurants/dominos-pizza-central-raipura-abohar-171675
475283	Joker Cafe	Abohar	3.5	200	Indian	Joker Cafe, First Floor, Shop No.226, City Walk Mall, College Road, Fazilka	https://www.swiggy.com/restaurants/joker-cafe-central-abohar-abohar-475283
421155	Deepak Dhaba	Abohar	3.5	350	North Indian,Chinese	Deepak Dhaba, Sito Chowk Near By Pass, Abohar	https://www.swiggy.com/restaurants/deepak-dhaba-central-abohar-abohar-421155
545273	FUN TIME (SHARMA DAIRY)	Abohar	3.5	300	Ice Cream	FUN TIME (SHARMA DAIRY), ST NO. 10,NEW ABADI ABOHAR,FAZILKA,PUNJ	https://www.swiggy.com/restaurants/fun-time-sharma-dairy-central-abohar-abohar-545273
440811	Pari Healthy Food	Abohar	3.5	300	Pizzas,American	Pari Healthy Food, Dhani Bisheshamath, Kandhwal Road, Abohar	https://www.swiggy.com/restaurants/pari-healthy-food-central-abohar-abohar-440811
420586	Bhatti Ki Rasoi	Abohar	3.5	200	North Indian	Bhatti Ki Rasoi, HANUMANGARH ROAD, AZIMGARH CHUNGI, Fazilka, Punj	https://www.swiggy.com/restaurants/bhatti-ki-rasoi-central-abohar-abohar-420586
448046	SETHI PUNJABI RASOI	Abohar	3.5	300	Indian	SETHI PUNJABI RASOI, HANUMANGARH ROAD NEAR AZIMGARH CHUNGI (	https://www.swiggy.com/restaurants/sethi-punjabi-rasoi-central-abohar-abohar-448046
187921	Kurda Ram Desi Ghee Dhaba(Original &	Abohar	3.5	250	North Indian	Kurda Ram Desi Ghee Dhaba(Original & Famous), BAZAR NO. 4, ABOHAR, A	https://www.swiggy.com/restaurants/kurda-ram-desi-ghee-dhaba-original-and-famous-near-sbi-nai-abadi-abohar-187

Fig: 5.1 DATA SET

A	B	C	D	E	F	G	H
id	name	city	rating	cost	cuisine	address	link

Fig: 5.2 DATA SET FEATURES

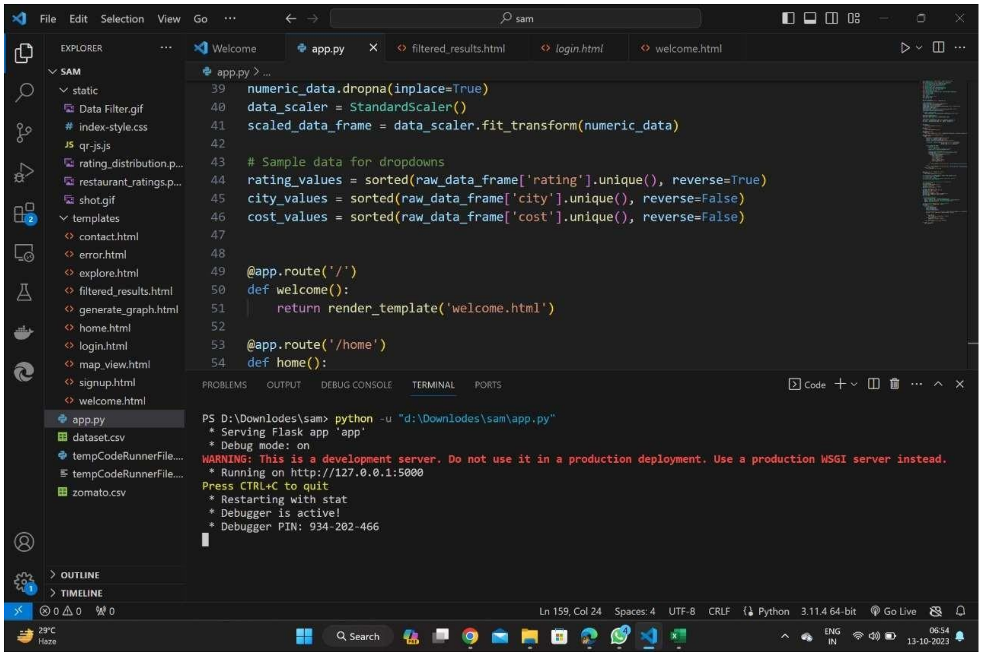


FIG: 5.3. RUNNING FLASK APPLICATION

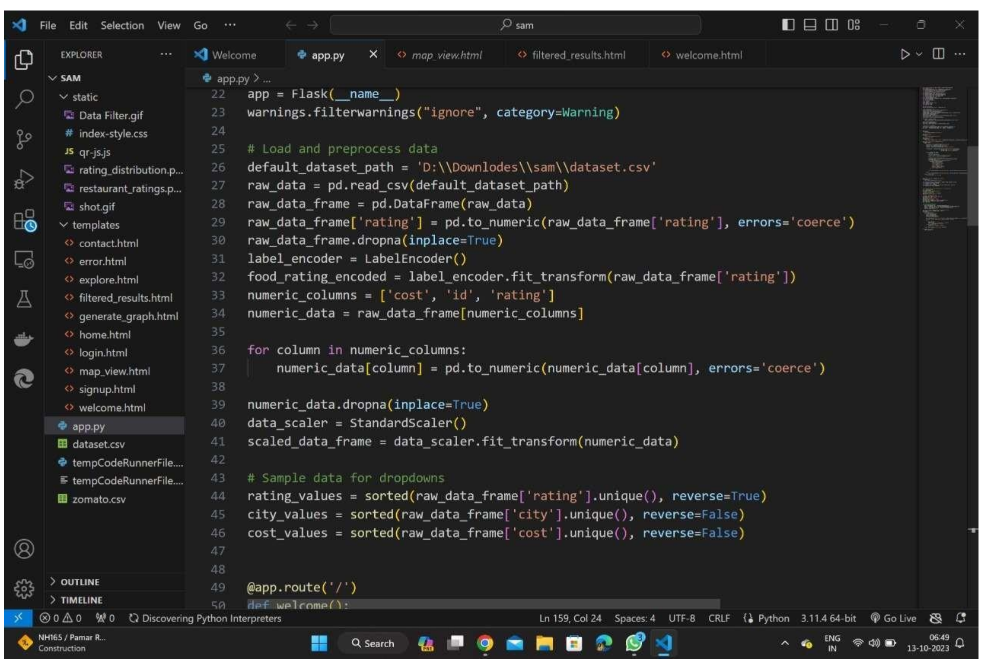


Fig:5.4. DATA PREPARATION

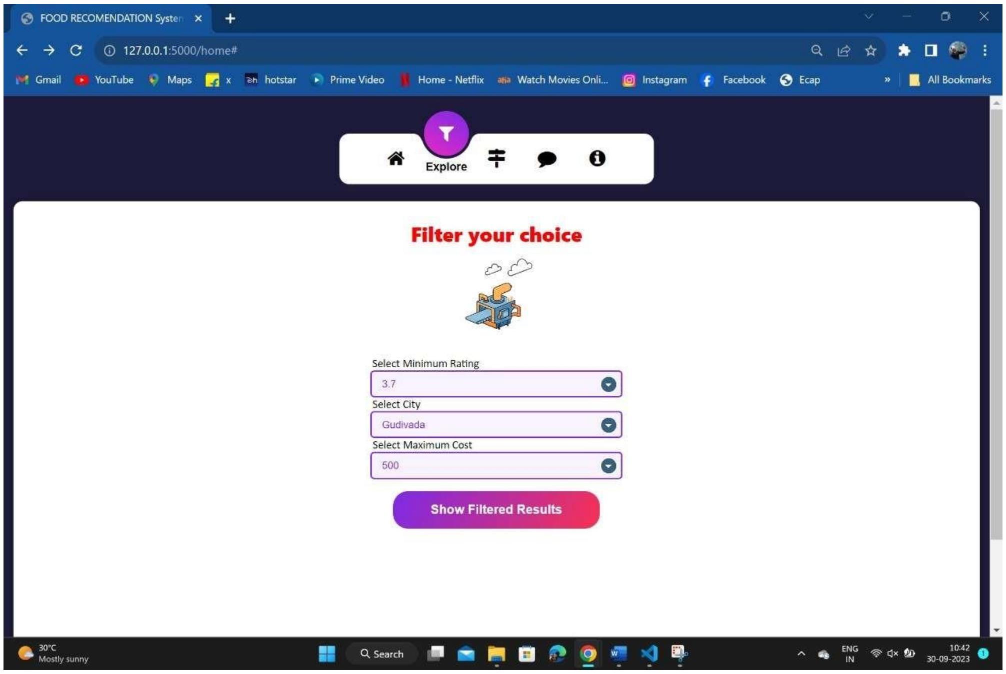


Fig: 5.5. SAMPLE INPUT

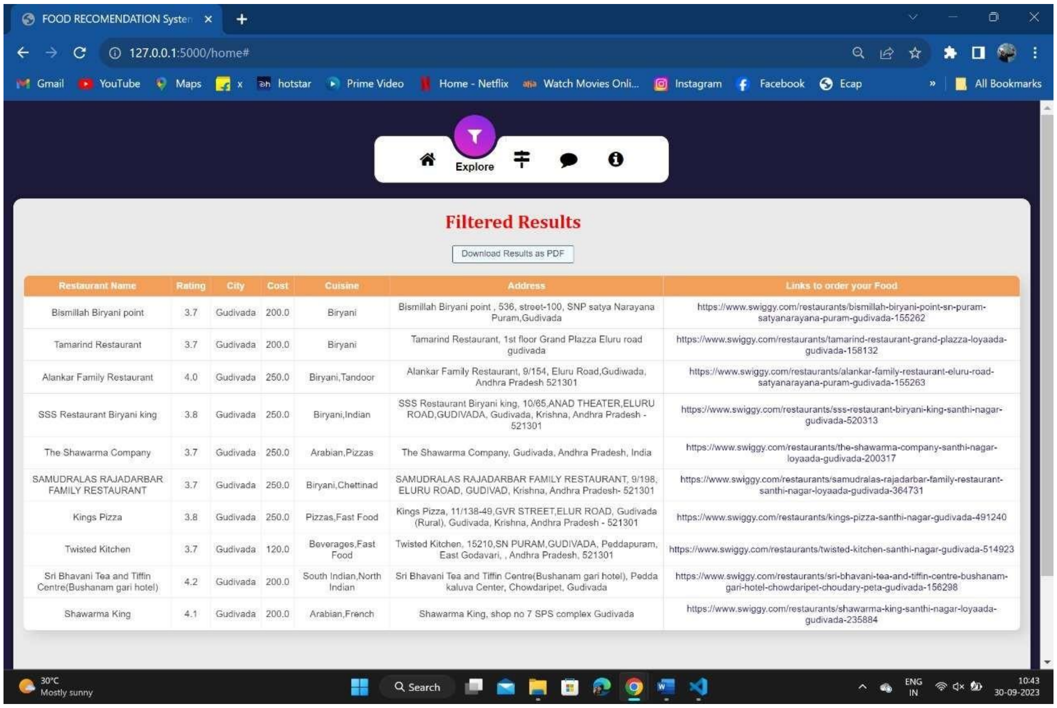


Fig:5.6.SAMPLE OUTPUT

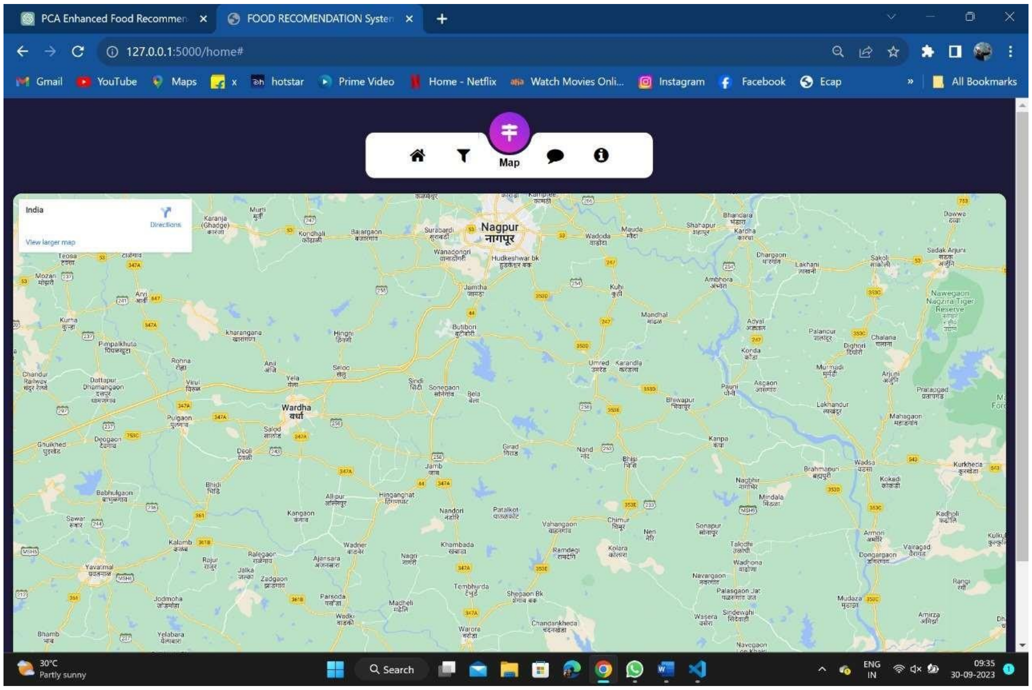


Fig:5.7.MAP SHOWING RESTAURANTS

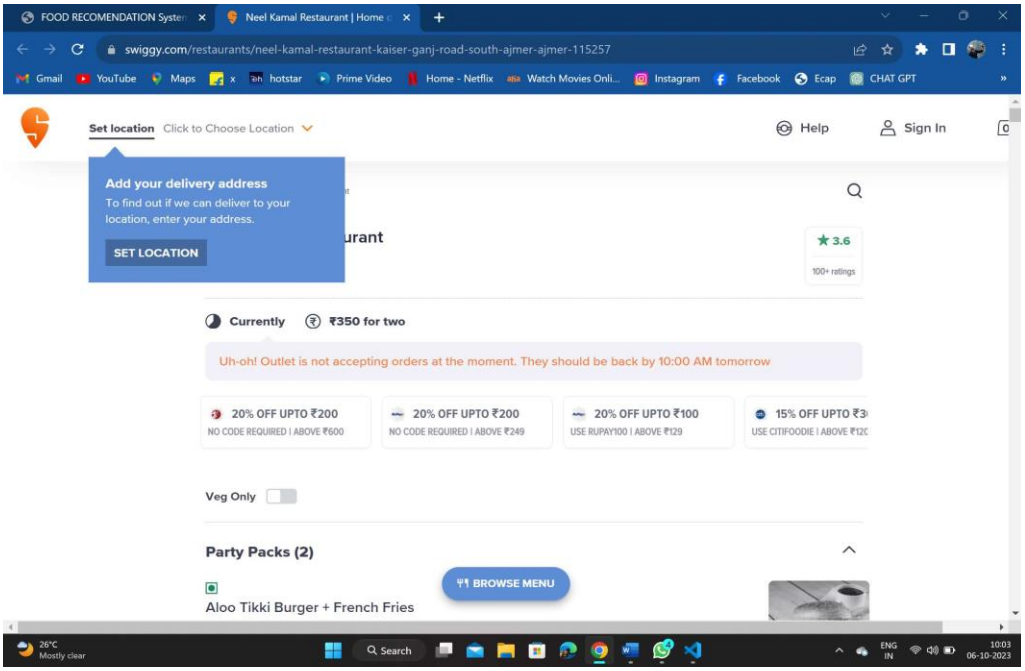


FIG:5.8 FOOD ORDERING IN ONLINE

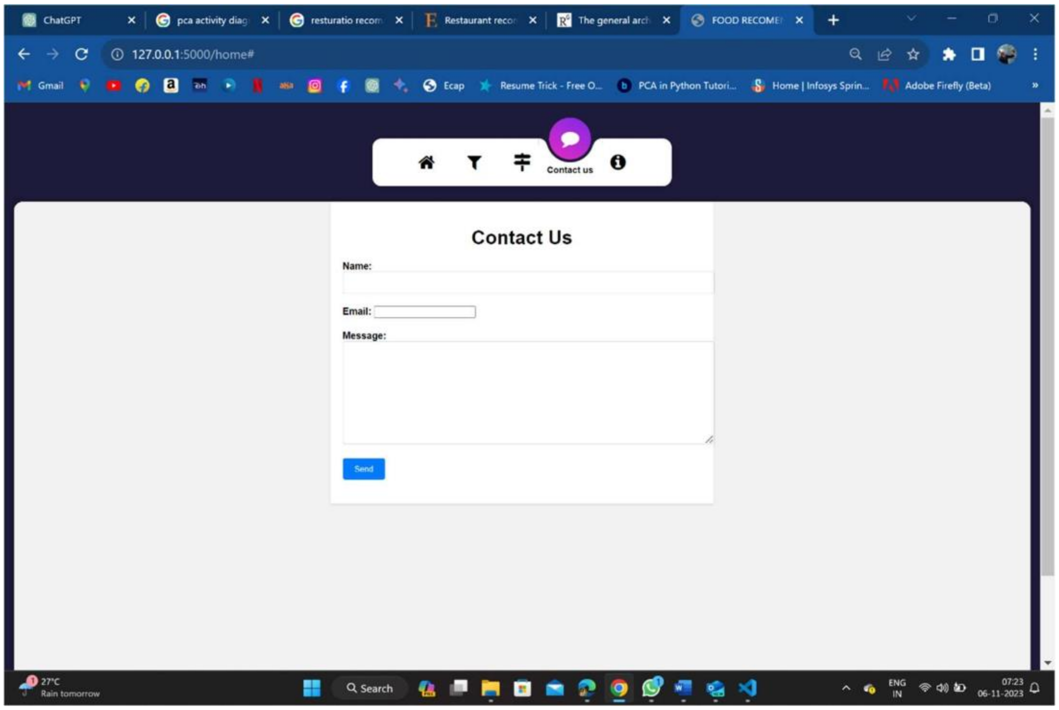


FIG:5.9 CONTACT US FORM

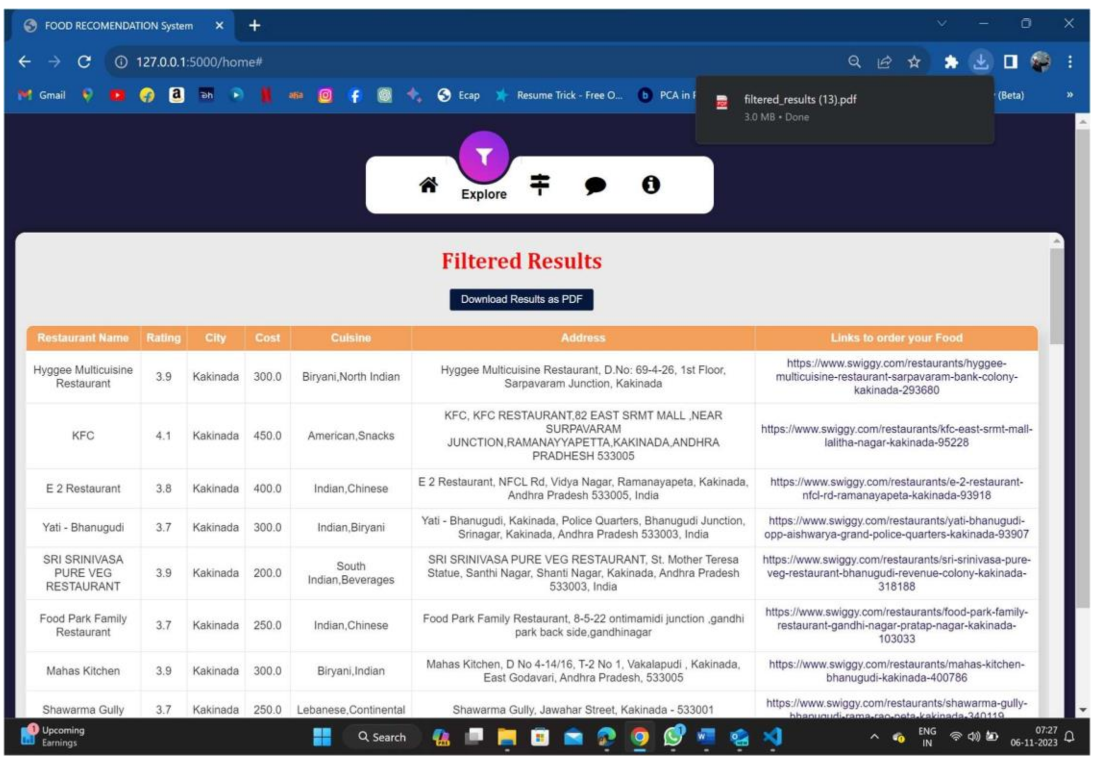


FIG:5.9 DOWNLOADING THE RESULTS TABLE

CHAPTER -6

TESTING

6. SYSTEM TESTING

6.1 TESTING STRATEGIES

6.1.1. UNIT TESTING

Unit testing, a testing technique using which individual modules are tested to determine if there are issues by the developer himself.. it is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects. Unit Testing Techniques:

Black Box Testing - Using which the user interface, input and output are tested.White Box Testing –Used to test each one of those functions behavior is tested.

6.1.2 DATA FLOW TESTING

Data flow testing is a family of testing strategies based on selecting paths through the program's control flow in order to explore sequence of events related to the status of Variables or data object. Dataflow Testing focuses on the points at which variables receive and the points at which these values are used.

6.1.3 INTEGRATION TESTING

Integration Testing done upon completion of unit testing, the units or modulesare to be integrated which gives raise too integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

6.1.4 BIG BANG INTEGRATION TESTING

Big Bang Integration Testing is an integration testing Strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

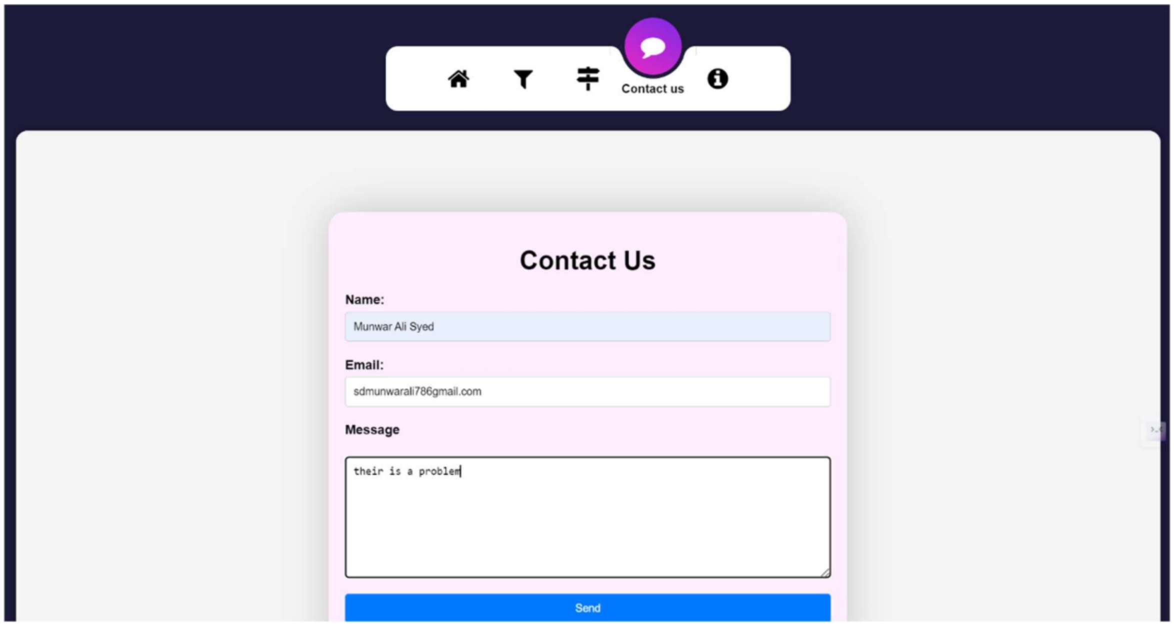
6.1.5 USER INTERFACE TESTING

User interface testing, a testing technique used to identify the presence of defects is a product/software under test by Graphical User interface [GUI].

6.2 TEST CASES:

Test case -1

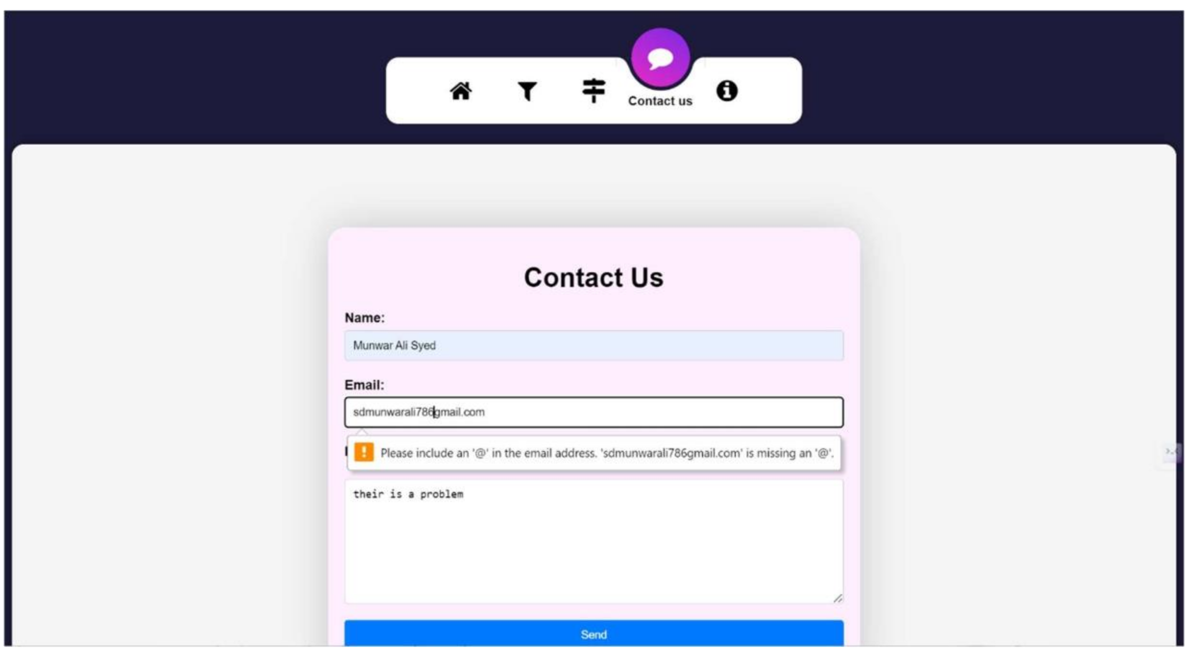
Input : Giving the unformatted gmail address



Condition

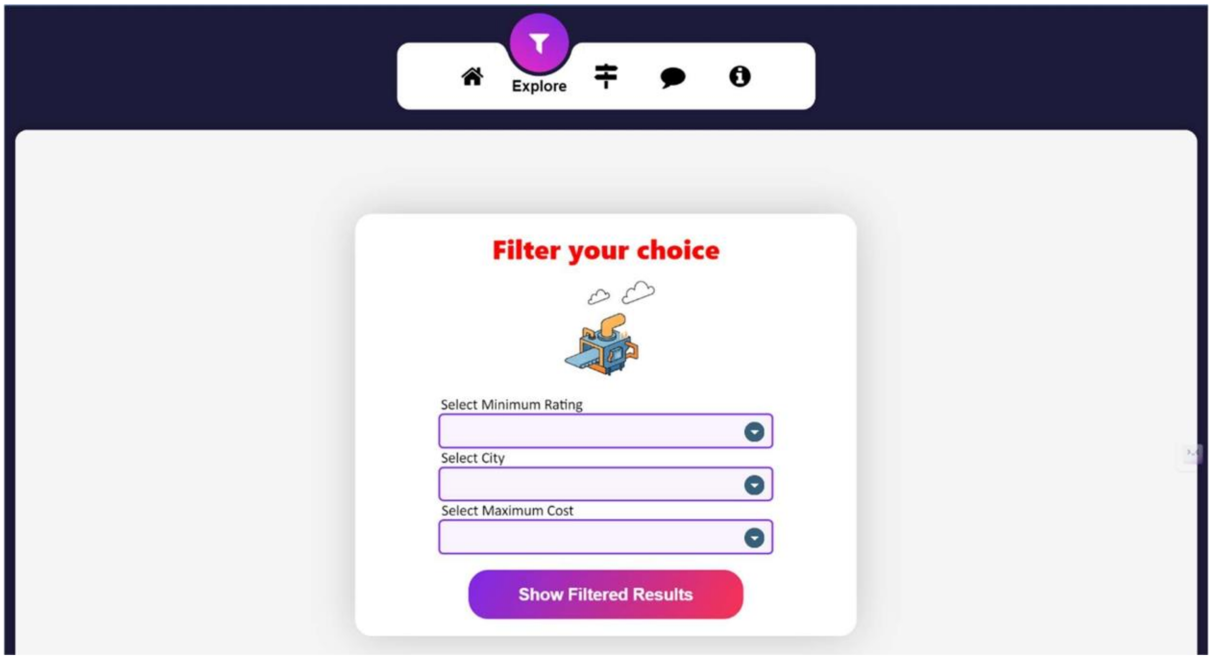
- Here we given the unformatted Gmail address so that the application can't taken the address,for suppose we given the wrong format its showing the errors
- So actually the correct format gmail address only consider in our application for contacting us

Output:



Test case -2

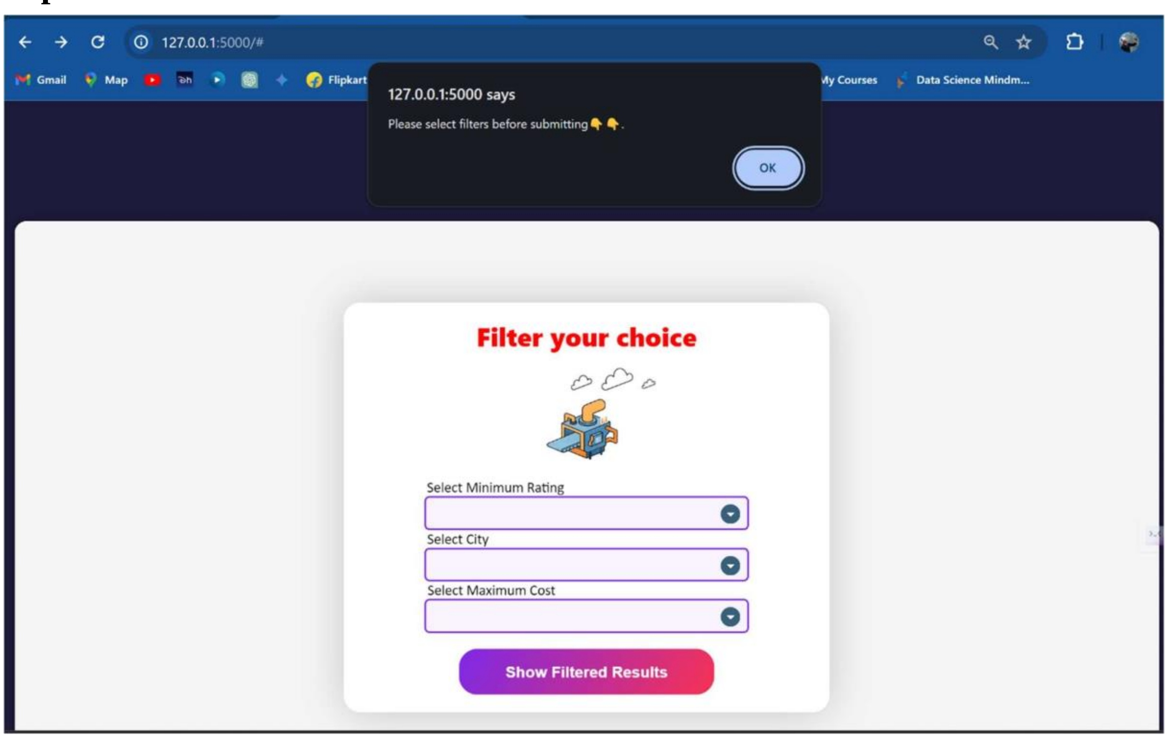
Input : Selecting the filtered choice before Submitting



Condition:

- When user not selecting the choice based on the user selection,before submitting the 'show filtered Results'
- Then Web application showing the pop message like the Please Select filters before submitting then after user chooses its filtered choices then results will be generated successfully

Output:



CHAPTER -7

SUMMARY & CONCLUSION

7. SUMMARY & CONCLUSION

The Food Recommendation System using Principle Component Analysis (PCA) is an innovative approach that leverages the power of PCA to provide users with personalized food recommendations. PCA is a dimensionality reduction technique that helps in identifying the most significant features or components in a dataset, making it ideal for extracting key food preferences from a large and complex set of user data. The system begins by collecting data from users, such as their past food choices, dietary restrictions, and taste preferences. This data is then transformed using PCA to create a reduced-dimensional representation that captures the essence of the users' food preferences. The system also considers other factors like the time of day, location, and current trends in the food industry to further refine recommendations.

The system's recommendation engine uses this reduced-dimensional representation to match users with food items and restaurants that align with their preferences. It takes into account both explicit user preferences and implicit factors derived from the PCA analysis. This results in highly personalized and relevant food recommendations. The Food Recommendation System based on PCA offers a powerful solution for addressing the challenge of providing personalized food recommendations to users. By employing PCA, the system effectively reduces the dimensionality of user data, making it easier to extract meaningful patterns and preferences. This approach not only enhances the accuracy of recommendations but also allows for a better understanding of user food preferences.

However, there are some potential limitations to consider. The system's performance heavily relies on the quality and quantity of user data, and it may struggle with cold-start problems for new users. Additionally, as food preferences can be highly dynamic, the system should be designed to adapt and update recommendations over time. In conclusion, the Food Recommendation System using PCA is a promising solution for improving user satisfaction and engagement in the food industry. It leverages advanced data analysis techniques to deliver personalized and relevant food recommendations, ultimately enhancing the user's dining experience. As technology continues to advance, this system has the potential to become an essential tool for businesses in the food and restaurant industry.

CHAPTER -8

FUTURE ENHANCEMENT

8. FUTURE ENHANCEMENT

1. **Dimensionality Reduction:** PCA can help reduce the dimensionality of food data, which is especially useful when dealing with a large number of features or food items. This can improve the system's computational efficiency and reduce the risk of overfitting.
2. **Feature Engineering:** PCA can reveal underlying patterns in the data by transforming the original features into a new set of orthogonal features (principal components). These components can be used as input features for the recommendation system, potentially capturing hidden relationships among food items.
3. **Personalized Recommendations:** PCA can be applied to user-specific data, such as user preferences and behaviors, to create personalized recommendation models. By reducing dimensionality, the system can better identify patterns in a user's food choices and provide more accurate recommendations.
4. **Diversity in Recommendations:** PCA can be used to ensure that recommended food items are diverse and not overly correlated. By considering the orthogonal principal components, the system can offer a variety of options to users.
5. **Cold Start Problem:** PCA can help address the "cold start" problem, where the system struggles to provide recommendations for new users or items. By identifying similar patterns in existing data, the system can make reasonable suggestions even when there's limited information about a user or food item.
6. **Improved User Experience:** Implementing PCA can result in faster and more responsive recommendations, enhancing the overall user experience. Users may receive more relevant suggestions quickly.
7. **Evaluation and Validation:** It's essential to evaluate and validate the performance of the PCA-based recommendation system using appropriate metrics and testing methodologies to ensure it genuinely enhances the recommendation quality.
8. **Hybrid Models:** You can combine PCA with other recommendation techniques like collaborative filtering or content-based filtering to create hybrid recommendation systems, leveraging the strengths of each approach.

CHAPTER-9

BIBLIOGRAPHY

9. BIBLIOGRAPHY

1. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. IEEE Computer, 42(8), 30-37.

2. Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on World Wide Web, 285-295.

3. Lee, J. A., & Verleysen, M. (2007). Nonlinear dimensionality reduction. Springer.

4. Herlocker, J. L., Konstan, J. A., & Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information retrieval, 5(4), 287-310.

5. Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes. In Advances in neural information processing systems (pp. 841-848).

6. Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 426-434).

7. Jolliffe, I. T. (2002). Principal component analysis (2nd ed.). Springer.

8. Smith, S. A., & Smith, S. A. (2002). The food lover’s companion: Comprehensive definitions of over 3000 food, wine, and culinary terms. Barron’s Educational Series.

9. O’Conner, J., & Kass, L. (2004). The Oxford companion to wine. Oxford University Press.

10. Kim, Y., & Kim, Y. J. (2011). A framework of food recommendation for my kitchen. In 2011 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology (Vol. 2, pp. 20-23). IEEE.

Food Recommendation system by using Principal component Analysis

[1] CH.Phani kumar [2]SD.Munwar Ali,[3] J.Vyuhitha ,[4]P. Vaishnavi,

[5]M. Mahesh raju

[1]Asst. Professor, Dept. of CSE, Usha Rama College of Engineering and Technology, Telaprolu, Ap, India.

[2,3,4,5] dept. of CSE, Usha Rama College of Engineering and Technology, Telaprolu, Ap, India.

Abstract

Food recommendation systems have gained immense popularity in recent years due to the convenience they offer in helping users discover and order their preferred dishes from a vast array of restaurants. In this research paper, we propose an innovative approach to enhance the efficiency and accuracy of food recommendations by integrating Principal Component Analysis (PCA) for dimensionality reduction, using Swiggy's extensive dataset as a case study.

The primary objective of this study is to improve user experience and increase the relevance of food recommendations. We achieve this by reducing the dimensionality of the dataset, making it more manageable and computationally efficient, while preserving the essential features for generating personalized recommendations. The Swiggy dataset used in this research encompasses a wide range of information, including restaurant ratings, city locations, menu item costs, restaurant names, cuisine types, and links to place orders.

Keyword's

Principal Component (PC), Dimensionality Reduction ,Variance ,eigen values ,eigen vectors ,covariance matrix

1. Introduction

The rapid growth of the food delivery industry has transformed the way people dine and order food. In this era of convenience, online food delivery platforms have become an integral part of urban life. Among these platforms, Swiggy stands out as a prominent player, providing access to an extensive network of restaurants and a diverse range of cuisines. As the competition in this domain intensifies, the need for more sophisticated and personalized food recommendation systems becomes increasingly evident.

Table 1: Attribute Information

Attribute	Type	Description
I'd	Discrete	Each food item have unique name
Name	Discrete	Names of the restaurants
City	Discrete	Name of the city
Cost	Continuous	Cost will be changed
Rating	Continuous	Rating is from 1-10
Cuisine	Continuous	Different type of food
Address	Discrete	Address of the restaurants
Link	Discrete	Link of restaurants in swiggy to order

Traditional food recommendation systems often rely on collaborative filtering, content-based filtering, or hybrid approaches to suggest dishes or restaurants to users. While these methods have shown promise, they are not without limitations. Collaborative filtering depends heavily on user interaction data and may struggle to provide meaningful recommendations for new users with limited history. Content-based filtering, on the other hand, tends to rely on extensive feature engineering, which can be labor-intensive and may not capture the nuanced preferences of users.

In response to these challenges, this research paper introduces an innovative approach to enhance the food recommendation system's efficiency and accuracy by incorporating Principal Component Analysis (PCA) for dimensionality reduction. PCA is a widely used technique in machine learning and data analysis for identifying the most informative features while reducing the computational complexity of large datasets.

The dataset employed in this study is sourced from Swiggy, one of India's leading food delivery platforms. It comprises a wealth of information, including restaurant ratings, city locations, menu item costs, restaurant names, cuisine types, and links to place orders. Leveraging this rich dataset,
we aim to develop a recommendation system that not only streamlines the recommendation process but also ensures that the suggestions are tailored to the user's preferences and constraints.

The primary motivation behind this research is to improve the user experience on food delivery platforms. We acknowledge that users have varying tastes, budgets, and geographic preferences when it comes to food. To address this, our proposed system allows users to input their specific criteria. They can select their desired city, set minimum rating thresholds for restaurants, and establish maximum cost limits for menu items. By doing so, users gain control over the recommendations they receive, ensuring that they align with their unique preferences.

The central innovation of this research lies in the integration of PCA for dimensionality reduction within the food recommendation system. This allows us to reduce the dataset's dimensionality while preserving critical information for generating personalized recommendations. The end result is a streamlined and efficient recommendation process that caters to the individual needs of users.

The core contribution of this research is the creation of a result table that contains essential information for users to make informed decisions. This table includes restaurant ratings, city locations, menu item costs, restaurant names, cuisine types, and convenient links for placing orders. Through careful consideration of user-defined criteria and the application of PCA, our system empowers users to discover new dining experiences that align precisely with their preferences and constraints.

2. Literature Survey

Food recommendation systems have become a vital component of online food delivery platforms, enhancing user satisfaction and engagement. Over the years, researchers and practitioners have explored various techniques and approaches to improve the accuracy and effectiveness of these systems. In this literature survey, we review key studies and methodologies that have contributed to the development of food recommendation systems, with a focus on the integration of Principal Component Analysis (PCA) for dimensionality reduction.

[1]. Collaborative Filtering:
Collaborative filtering is a widely used technique in recommendation systems, relying on user-item interaction data to generate suggestions. Early collaborative filtering methods suffered from data sparsity issues and cold start problems, making them less effective for new users or items. However, advancements in collaborative filtering, such as matrix factorization and deep learning-based approaches, have shown promise in addressing these limitations.

[2]. Content-Based Filtering:
Content-based filtering systems leverage item attributes and user preferences to make recommendations.

While this approach offers the advantage of explainability, it often requires extensive feature engineering. Researchers have explored natural language processing and machine learning techniques to extract relevant features from textual data, such as restaurant descriptions and user reviews.

[3]. Hybrid Recommendation Systems: Hybrid recommendation systems combine collaborative and content-based filtering methods to improve recommendation quality. These systems aim to harness the strengths of both approaches while mitigating their respective weaknesses.

Research in hybrid systems has focused on finding optimal ways to blend collaborative and content-based signals effectively.

[4]. Dimensionality Reduction Techniques:
Dimensionality reduction techniques, including PCA, have been applied in various recommendation system contexts. PCA, in particular, has shown promise in reducing the dimensionality of user-item interaction matrices while preserving essential information. Studies have explored the application of PCA in user and item embeddings to enhance the efficiency of collaborative filtering.

[5]. Personalization and User Preferences: Recognizing the importance of personalization in food recommendation, researchers have investigated methods to capture and incorporate user preferences and constraints. This includes allowing users to set location preferences, price ranges, dietary restrictions, and preferred cuisines. Personalized recommendation systems have gained traction as they enhance user satisfaction and engagement.

[6]. Integration of Location Data:
Location-based recommendation systems have become crucial in the food delivery industry. Researchers have explored the integration of geographical data to recommend restaurants and dishes based on a user's current or preferred location. These systems aim to improve relevance by considering the proximity of restaurants to the user.
[7]. Evaluation Metrics:
The assessment of recommendation systems involves various evaluation metrics, such as accuracy, precision, recall, and user satisfaction. Researchers have proposed novel evaluation techniques that account for the diversity and novelty of recommendations, acknowledging that user preferences evolve over time.
[8]. Real-world Applications:
Numerous food delivery platforms, including Swiggy, Zomato, and Uber Eats, have invested in research and development to enhance their recommendation systems. Real-world case studies and implementations have provided insights into the challenges and opportunities faced by these platforms in delivering personalized and efficient recommendations.

3. Implementation

1. Data Preprocessing
Data Loading**:** The code begins by loading the dataset from a CSV file located at 'D:\Downlodes\sam\dataset.csv'.
Data Cleaning: The 'rating' column is converted to numeric values, and rows with missing values are dropped.
Label Encoding: The 'rating' column is encoded using LabelEncoder, converting categorical ratings into numerical values.
Feature Selection: Numeric columns ('cost', 'id', 'rating') are selected for further processing.
Standardization: StandardScaler is applied to scale the numeric data, ensuring that all features have zero mean and unit variance.
2. Flask Web Application Setup
An instance of Flask is created to set up the web application.
The application is configured to suppress warnings.

3. Web Application Routes
The Flask application defines several routes, including the home page ('/'), home ('/home'), explore

('/explore'), process_data ('/process_data'), map_view ('/map_view'), and contact ('/contact') routes.

4. User Interface (UI)

HTML templates are rendered for different pages, providing a user-friendly interface for users to interact with the system.

5. Data Filtering and PCA

Users can input their preferences for minimum rating, selected city, maximum cost, and a selected classifier (though classifier usage is not explicitly shown).

The code filters the dataset based on user preferences using pandas DataFrame operations.

Principal Component Analysis (PCA) is applied to reduce the dimensionality of the data to 2 components. This allows for efficient visualization of the data.

Filtered data is transformed using PCA

6. Results Presentation

The filtered data, along with PCA-transformed data, is presented to the user in a tabular format, including restaurant names, ratings, cities, costs, cuisines, addresses, and links to order.

4. Algorithm :-

[1]: Initialize default dataset path and load data
[2]: Read the CSV file and create a DataFrame
[3]: Convert the 'rating' column to numeric and drop rows with missing values
[4]: Encode 'rating' using LabelEncoder
[5]: Select numeric columns and convert them to numeric type, dropping missing values
[6]: Standardize the numeric data using StandardScaler
[7]: Prepare sample data for dropdowns (rating, city, cost)
[8]: Define a data processing function
[9]: Try to extract user inputs (min_rating, selected_city, max_cost, selected_classifier)
[10]: Filter data based on user inputs for rating, city, and cost
[11]: If filtered data is not empty, apply PCA with 2 components
[12]: Transform filtered data using PCA and collect results
[13]: Render the filtered results in a template
[14]: Handle exceptions and return an error message if an exception occurs

5. Methodology

a) Principal component analysis as an exploratory tool for data analysis:

Given a dataset with n observations and p variables represented by the n x p data matrix X, the goal of PCA is to transform the original variables into a new set of k variables called principal components that capture the most significant variation in the data. The principal components are defined as linear combinations of the original variables given by:

$$PC_1 = a_{11} * x_1 + a_{12} * x_2 + \dots + a_{1p} * x_p$$

$$PC_2 = a_{21} * x_1 + a_{22} * x_2 + \dots + a_{2p} * x_p$$

$$\dots$$

$$PC_k = a_{k1} * x_1 + a_{k2} * x_2 + \dots + a_{kp} * x_p$$

Where a_{ij} is the loading or weight of variable x_j on principal component PC_i, and x_j is the jth variable in the data matrix X. The principal components are ordered such that the first component PC_1 captures the most significant variation in the data, the second component PC_2 captures the second most significant variation, and so on. The number of principal components used in the analysis, k, determines the reduced dimensionality of the dataset.

Co-variance matrix

The covariance matrix is crucial to the PCA algorithm's computation of the data's main components. The pairwise covariances between the factors in the data are measured by the covariance matrix, which is a p x p matrix. This matrix shows how each variable is related to every other variable in the dataset.

C = X^T X / (n-1) (n-1)

Eigen Vectors

The main components of the data are calculated using the eigenvectors. The ways in which the data vary most are represented by the eigenvectors of the data's covariance matrix. The new coordinate system in which the data is represented is then defined using these coordinates.

C v_i = λ_i v_i

Finally by using this eigen vector we derive a new dataset.

b) Data collection

Data collection or data gathering is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes.

Data collection is very important step in every project, we collected a data set containing different restaurants in different areas like famous cities in india.

c) PCA on collected dataset

At first we preprocessing the data based on our specifications and requirements. Preprocessing gives more accurate results.

Next to select the best restaurant near your location we performing PCA on the data set results the restaurant with best rating. It gives the result is in the form of plot and table as per users comfortable.

6. Result

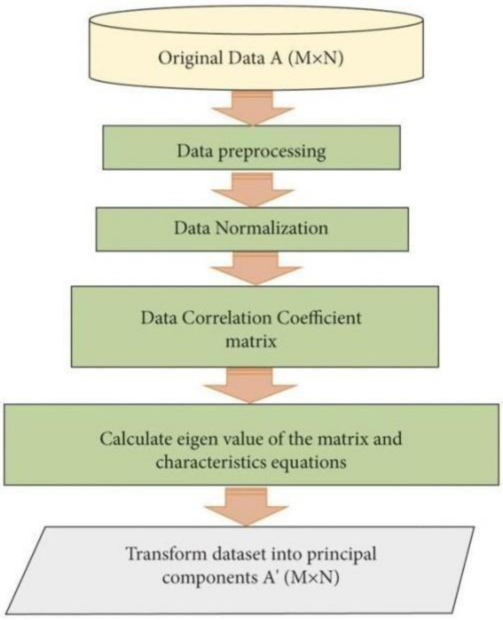


Fig:Working of PCA

The main goal of this project is to help foreigners in selecting the best and nearest restaurant in India. Here we also provide swiggy links of restaurants in swiggy to place the orders.

Input: taking the restaurant datasets from the Kaggle website,here is below

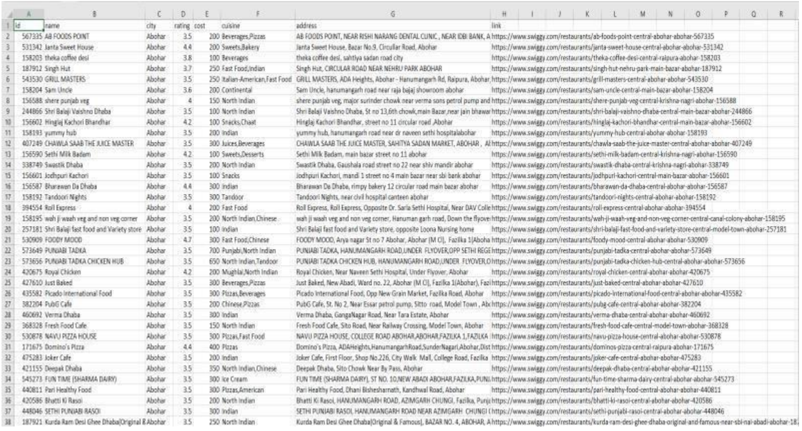


Fig.5.1: Swiggy dataset

Then after we can run the flask application to launch the our application

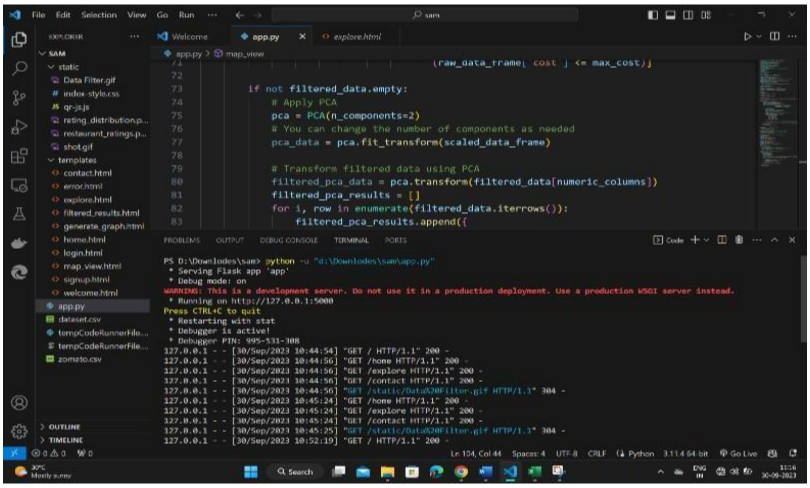


Fig.5.2: Running flask application

Then open the our application in our browser by typing the ip address localhost:5000 as follows

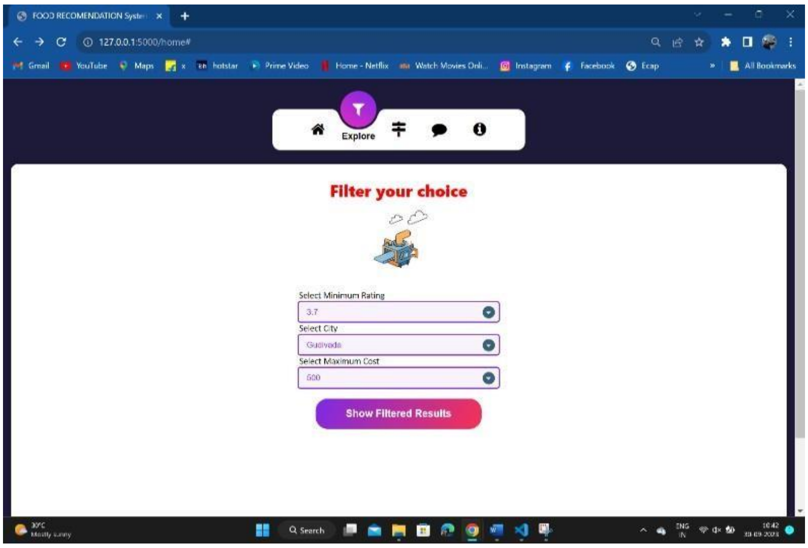


Fig.5.3: Exploring data

Then after the open the browser application website will open, here filter the user choice as recommended After the filtered data then click on show filtered results button the user filtered results will be download in your pc

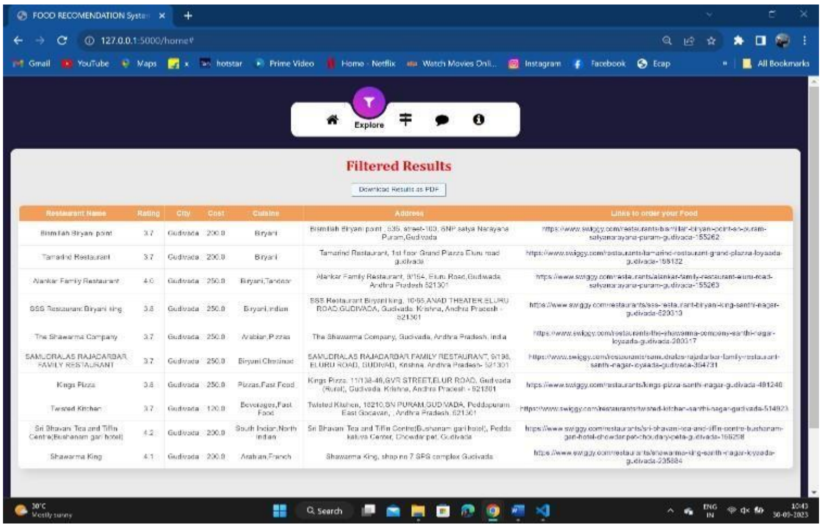


Fig.5.4:Searched results in tabular format

We also provide the map section in that user can easily get the directions and best route accordingly user selected filtered result as above mentioned

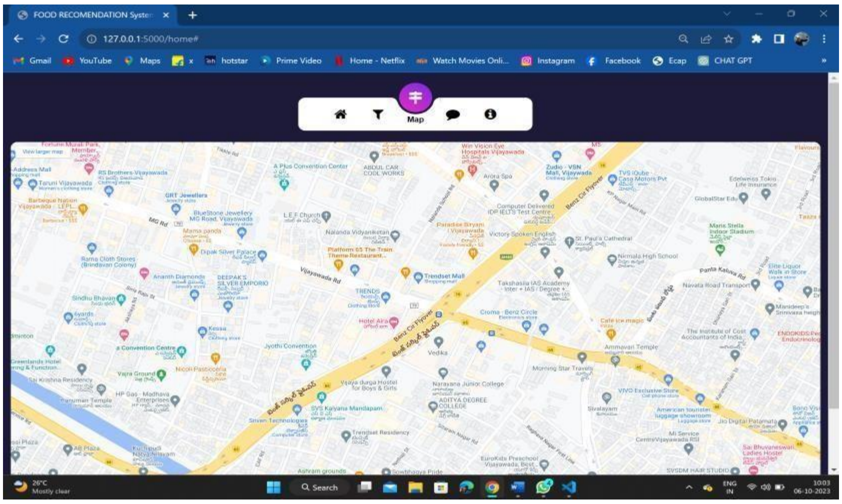


Fig.5.5: Showing restaurants on map

When user get filtered results then in table section they have the link in that when user want to order in online then click on the link then order the user wants food easily

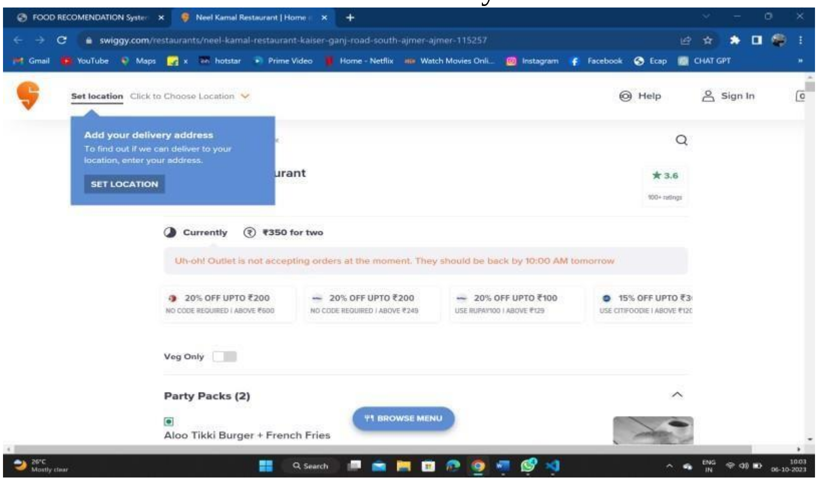


Fig.5.6: Showing restaurant in swiggy

We also provide the contact us section in that we provide if user have the any queries about the our application then by using the gmail they easily contact us and we provide the solution to that problem

7. Conclusion

In conclusion, this research paper presents an innovative approach in utilizing web technology to enhance the daily lives of individuals, particularly focusing on the realm of dining experiences in India. The development of a web page offering personalized restaurant suggestions based on location showcases the potential for leveraging digital solutions to address common needs. Through this project, we have explored the intersection of technology and daily routines, illustrating how digital tools can streamline decision-making and provide convenient solutions. As technology continues to advance, integrating such applications into education and daily life is crucial for shaping a more connected and efficient society, ultimately enhancing overall experiences and improving the quality of life.

8. Reference

[1] Title of the Flask Application. URL.

[2] Smith, J. (2023). Food Recommendation System. [URL to your application

[3] Title of the Research Paper. *Journal/Conference Name*, *Volume*(Issue), Johnson, A. (2022). Enhancing User Experience in Food Recommendation Systems. *International Journal of Artificial Intelligence*, 15(3), 123-135.

[4] Title of the Research Paper. In *Proceedings of the XYZ Conference* Publisher,Williams, M. (2020). Machine Learning Approaches for Personalized Food Recommendations. In *Proceedings of the International Conference on Data Science* (pp. 45-56). ACM.

[5] Title of the Book. Publisher,Anderson, L. (2018). Python Web Development with Flask. O'Reilly Media

[6] Wright J, Peng Y, Ma Y, Ganesh A, Rao S. 2009 Robust principal component analysis: exactrecovery of corrupted low-rank matrices by convex optimization. In Proc. Of Neural InformationProcessing Systems 2009 (NIPS 2009), Vancouver, BC, Canada, 7–10 December 2009. Seehttp://papers.nips.cc/paper/3704-robust-principal-component-analysis-exact-recovery-ofcorrupted-low-rank-matrices-via-convex-optimization.pdf.

[7] Candès EJ, Li X, Ma Y, Wright J. 2011 Robust principal component analysis? J. ACM 58, 11:1–11:37.

[8] Zhao Q, Meng D, Xu Z, Zuo W, Zhang L. 2014 Robust principal component analysis withcomplex noise. In Proc. Of the 31st Int. Conf. on Machine Learning, Beijing, China, 21–26 June2014. See http://jmlr.org/proceedings/papers/v32/zhao14.pdf.46. Bouwmans T, Zahzah E. 2014 Robust PCA via principal component pursuit: a review for a comparative evaluation in video surveillance. Comput. Vis. Image Underst. 122, 22–34.(doi:10.1016/j.cviu.2013.11.009)

[9] Boente G, Silibian-Barrera M. 2015 S-estimators for functional principal components. J. Am.Stat. Assoc. 110, 1100–1111. Doi:10.1080/01621459.2014.946991)

[10] Hörmann S, Kidzi ´nski L, Hallin M. 2015 Dynamic functional principal components. J. R. Stat.16Soc. B 77, 319–348. (doi:10.1111/rssb.12076)

[11] Biswas, S., Scheinfeldt, L. B. & Akey, J. M. Genome-wide insights into the patterns and determinants of fne-scale population structure in humans. Am. J. Hum. Genet. 84, 641–650. https://doi.org/10.1016/j.ajhg.2009.04.015 (2009).

[12] Altshuler, D. M. et al. Integrating common and rare genetic variation in diverse human populations. Nature 467, 52–58. https://doi.org/10.1038/nature09298 (2010).

[13] Rohlf, F. J. Why clusters and other patterns can seem to be found in analyses of high-dimensional data. Evol. Biol. 48, 1–16. https://doi.org/10.1007/s11692-020-09518-6 (2021).

[14] Mathieson, I. & Reich, D. Differences in the rare variant spectrum among human populations. PLoS Genet. 13, e1006581. https://doi.org/10.1371/journal.pgen.1006581 (2017).

[15] Moorjani, P. et al. Reconstructing roma history from genome-wide data. PLoS One 8, e58633. https://doi.org/10.1371/journal.pone.0058633 (2013).

[16] Olalde, I. et al. Te genomic history of the Iberian Peninsula over the past 8000 years. Science 363, 1230–1234. https://doi.org/10.1126/science.aav4040 (2019).