Restaurant Recommendation System

Final Project Report

Submitted To:

Smart Internz

Applied Data Science – Guided Project

Submitted By:

Simran Nayak

D.Y. Patil Agriculture and Technical University, Talsande

Submission Date:

20 June 2025

Abstract

This project presents an intelligent restaurant recommendation platform that personalizes dining suggestions based on individual user tastes, location, and prior activity. It evaluates multiple factors like cuisine, cost, distance, and reviews to suggest ideal food spots. Employing machine learning models—especially hybrid systems combining collaborative and content-based filtering—this solution aims to enhance decision-making and improve the overall dining experience by offering precise, location-tailored recommendations.

TABLE OF CONTENTS

1. Introduction

1.1 Project Overview

The **Restaurant Recommendation System** is a personalized digital assistant that helps users discover suitable eateries that align with their tastes and situational preferences. With the explosion of food outlets and platforms like Zomato, Swiggy, and Google Reviews, consumers often face choice paralysis. Our system tackles this challenge using a **hybrid recommendation model** that blends user behavior analysis (collaborative filtering) with item-based attributes (content filtering), and also considers geographical context through location data.

This multi-layered approach ensures recommendations are not only tailored but also relevant to the user's current setting. Whether someone craves spicy street food or fine dining near a specific location, the system adapts accordingly. Furthermore, the solution is capable of **learning and evolving** with every interaction using feedback loops to refine its accuracy over time.

1.2 Project Objectives

1. Build a smart recommendation engine that prioritizes individual food preferences—cuisine type, budget, ambiance, and dietary needs.
2. Implement collaborative and content-based machine learning algorithms to detect behavioral and contextual patterns.
3. Embed location-awareness using GPS data or manually entered location preferences.
4. Analyze user reviews and third-party ratings to improve credibility and suggestion quality.
5. Create an intuitive user interface that allows smooth input and result visualization.
6. Incorporate a feedback collection mechanism for continuous learning and better personalization.
7. Design the system to be modular, scalable, and ready for cross-region deployment or app integration.

2. Project Initialization and Planning Phase

2.1 Problem Definition

User-Centered Problem Scenarios

PS No.	I am a...	I'm trying to...	But...	Because...	Which makes me feel...
PS-1	Tourist in a new city	Discover local, trustworthy food spots	I don't know the place well	No guidance on regional specialties or ratings	Confused and unsure where to eat
PS-2	Strict vegetarian	Find purely vegetarian restaurants	Most apps mix veg and non-veg	I need clear filtering	Frustrated and excluded
PS-3	Small restaurant owner	Gain visibility on food apps	My place rarely appears in suggestions	Platforms prioritize larger brands	Disheartened and overlooked
PS-4	Student on a tight budget	Eat out without breaking the bank	Expensive places are always listed first	Filters don't prioritize budget	Pressured and discouraged
PS-5	Regular delivery app user	Receive relevant meal suggestions	It doesn't learn from my history	Recommendations repeat or miss the mark	Bored and unimpressed
PS-6	Parent with young kids	Choose family-safe dining options	No way to filter child-friendly restaurants	Safety and cleanliness are unclear	Worried and uneasy
PS-7	New town resident	Explore diverse, cultural cuisines	I miss out on hidden gems	The system lacks cultural tagging	Disconnected and bored
PS-8	Health-focused eater	Locate nutritious	Menus lack calorie or	I can't make informed decisions	Detached from health goals

		dining options	ingredient info		
PS-9	Food platform analyst	Monitor food safety in listings	Platforms don’t verify ingredients	No AI-driven verification systems exist	Concerned about public trust
PS-10	Data scientist	Study food trends via reviews	Data is unstructured or incomplete	Hard to train models or draw insights	Blocked and limited
PS-11	Foodie traveler	Find must-try restaurants in new cities	Results are generic and far off	It lacks personalization	Let down and lost
PS-12	Restaurant owner	Manage and improve my profile	I can’t update outdated reviews	Platforms limit my access	Powerless and misrepresented

2.2 Project Proposal (Proposed Solution)

Objective

To develop a robust, intelligent restaurant recommendation system that delivers personalized dining suggestions by analyzing user preferences, dietary constraints, budget, and geolocation data. The system is designed to enhance decision-making efficiency and improve user satisfaction by providing relevant and adaptive restaurant recommendations.

Scope

The project targets diverse user groups—ranging from tourists and students to health-conscious individuals and restaurant owners. The system will operate across multiple regions and will integrate both static and dynamic data, including real-time user input, public reviews, and platform APIs. It aims to be scalable for future integration into third-party delivery or travel platforms.

Problem Statement

Modern users often struggle to identify restaurants that align with their specific needs, such as dietary requirements, cost preferences, or location constraints. Existing recommendation platforms provide generalized suggestions that fail to adapt to individual behavior or contextual factors. As a result, users face decision fatigue, repeat visits to familiar places, and miss out on better-suited dining experiences.

Impact

Addressing this issue will lead to several positive outcomes:

- Improved personalization of restaurant discovery
- Enhanced support for small and local businesses
- Increased efficiency in decision-making
- Better alignment with user lifestyle, budget, and health goals

Proposed Solution

The proposed system employs a hybrid machine learning approach that integrates the following components:

- **Collaborative Filtering**: Analyzes user behavior and preferences to recommend restaurants based on the behavior of similar users.
- **Content-Based Filtering**: Uses restaurant-specific attributes (e.g., cuisine, cost, ambiance, ratings) to match user-stated preferences.
- **Geolocation Integration**: Incorporates real-time GPS or IP-based location data to ensure suggestions are contextually relevant.
- **Feedback Mechanism**: Continuously adapts to user behavior and satisfaction ratings to improve the recommendation quality over time.

This architecture allows for dynamic, location-aware, and preference-sensitive suggestions that evolve with continued usage.

Key Features

- Personalized recommendation engine
- Real-time data processing and filtering
- Review and rating aggregation
- Dietary and budget-based filtering
- Location-aware search functionality
- Scalable and modular system architecture

Resource Requirements

Resource Type	Description	Specification
Hardware	Processing Power	8-core CPU with optional GPU support (2x NVIDIA V100)
	Memory	Minimum 8 GB RAM
	Storage	1 TB SSD for persistent storage of user profiles and restaurant metadata
Software	Programming Language	Python
	Frameworks	Flask (Web), Scikit-learn, TensorFlow
	Libraries	Pandas, NumPy, BeautifulSoup (Web Scraping), NLTK (Text Processing)
	Development Environment	Jupyter Notebook, Git for version control
Data	Format	CSV (structured datasets), HTML/Text (scraped reviews)
	Source	Public APIs (e.g., Yelp, Zomato), user-generated content, crowdsourced review platforms
	Volume	Approx. 50,000 to 100,000 records, scalable based on usage growth

2.3 Initial Project Planning

Product Backlog, Sprint Schedule, and Estimation

The development process followed an Agile methodology with two initial sprints, aimed at quickly iterating through core functionalities and enhancements. Each user story was carefully planned and assigned based on complexity, business value, and development effort.

Sprint	Functional Requirement (Epic)	User Story No.	User Story Description	Story Points	Priority	Sprint Start Date	Sprint End Date
Sprint 1	User Preferences Input	USN-1	As a user, I want to enter my food or restaurant preferences so that I can receive personalized suggestions.	2	High	01 June 2025	02 June 2025
Sprint 1	Recommendation Engine	USN-2	As a user, I want to receive restaurant recommendations based on my input.	3	High	02 June 2025	02 June 2025
Sprint 2	Review & Rating Integration	USN-3	As a user, I want to view restaurant ratings and reviews fetched from available data sources.	2	Medium	03 June 2025	04 June 2025

Sprint 2	UI/UX Enhancement	USN-4	As a user, I want the results displayed in a clean interface with sorting and filtering capabilities.	2	Medium	04 June 2025	05 June 2025

3. Data Collection and Preprocessing Phase

3.1 Data Collection Plan and Raw Data Sources Identified

Project Overview

The objective of this project is to create a restaurant recommendation system capable of delivering tailored suggestions by analyzing various inputs such as cuisine preferences, location, pricing, and user reviews. This system will rely on structured restaurant metadata and user-generated data to train the model and improve recommendation quality.

Data Collection Plan

To develop the recommendation engine, a real-world dataset was procured from Kaggle containing detailed information on restaurants in Bangalore. This dataset includes attributes such as restaurant names, locations, cuisine types, customer ratings, and pricing, which are crucial for both exploratory data analysis and machine learning model training.

The selection of this dataset was based on its richness, structure, and public accessibility. It serves as a foundational resource for understanding dining patterns and preferences in an urban setting.

Raw Data Source Overview

Source Name	Description	Location/URL	Format	File Size	Access Level
SmartInternz Dataset	Restaurant-level structured metadata including name, location, cuisines, rating, delivery, and cost-related features.	Zomato Bangalore Dataset – Kaggle	CSV	~93 MB	Public

3.2 Data Quality Report

The dataset underwent a comprehensive quality assessment to identify and resolve issues such as missing fields, duplication, inconsistent formatting, and sparsity. The table below outlines the problems encountered and the respective remediation strategies:

Data Source	Identified Issue	Severity	Resolution Strategy
Restaurant Metadata	Missing values in fields like restaurant name, location, or user rating	Moderate	Use data imputation techniques (e.g., mean, mode, or default value substitution). For rows with critical missing values, remove records entirely.
User Reviews	Duplicate review entries and timestamp inconsistencies	Low	Eliminate duplicates using `drop_duplicates()` in Pandas or `DISTINCT` in SQL. Normalize date/time fields using libraries such as `pandas.to_datetime()`.
Location Fields	Inconsistent city/location entries (e.g., "Mumbai" vs. "BOM")	Moderate	Perform string cleaning using regex and mapping functions to standardize location names.
User Preference Data	Sparse or insufficient user interaction history	High	Implement fallback strategies such as popularity-based or content-based filtering in the absence of collaborative data. Default suggestions may also be shown based on trending or high-rated restaurants.

3.3 Data Preprocessing

Section Description

Data Overview

The dataset used in this project contains structured information about restaurants, including attributes such as name, location, average cost, user reviews, ratings, and types of cuisine. This data was extracted from a public Zomato dataset and prepared for use in a content-based restaurant recommendation engine.

Before feeding the data into machine learning models, several preprocessing steps were applied to enhance data quality, reduce noise, and ensure consistency across the dataset. These transformations are critical for improving model performance, especially in the context of natural language processing (NLP) and numerical feature analysis.

Preprocessing Techniques Applied

Preprocessing Step	Applicability	Description
Resizing	Not Applicable	This step is relevant for image data. As the dataset is text and tabular, resizing was not performed.
Normalization	Applied	Numerical fields such as user ratings and average cost were normalized using `MinMaxScaler` to a uniform range (e.g., 0–1 or 1–5). Text data was converted to lowercase, and excessive whitespace and punctuation were removed.
Data Augmentation	Not Applicable	Not required, as augmentation is typically used in image or speech datasets to artificially expand the dataset.
Denoising	Applied	Text reviews and metadata were cleaned by removing newline characters, special symbols, and unnecessary punctuation. Stopwords were filtered to focus on sentiment-bearing words.
Edge Detection	Not Applicable	This technique is exclusive to image data and was not

		relevant to this project.
Color Space Conversion	Not Applicable	As no image processing was involved, color space transformation was not performed.
Image Cropping	Not Applicable	No image-based features were used; therefore, cropping was unnecessary.
Batch Normalization	Not Applicable	This is a neural network training technique primarily used in image-based CNN models. It was not applied here.

```python
# Importing necessary libraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import re

# Load dataset
df = pd.read_csv("zomato_bangalore.csv")

# Normalize ratings
scaler = MinMaxScaler(feature_range=(1, 5))
df['normalized_rating'] = scaler.fit_transform(df[['aggregate_rating']])

# Clean textual data (e.g., cuisines, restaurant names)
def clean_text(text):
    text = str(text).lower()
    text = re.sub(r'[^\w\s]', '', text)
    text = re.sub(r'\s+', ' ', text)
    return text.strip()

df['cuisines'] = df['cuisines'].apply(clean_text)
df['restaurant_name'] = df['restaurant_name'].apply(clean_text)
```

4. Model Development Phase

4.1 Model Selection Report

Model Evaluation and Selection Overview

In the development of the restaurant recommendation system, several machine learning techniques were considered and evaluated based on their effectiveness, complexity, scalability, and suitability for the data characteristics. The following section outlines the models studied and the rationale for selecting the final approach.

4.1 Model Selection Report

Model	Description	Advantages	Limitations
Content-Based Filtering	Recommends restaurants by matching user preferences (e.g., cuisine, price, dietary needs) to restaurant attributes.	- Personalized to user's specific interests - Independent of other users' data	- Limited diversity in suggestions - Cold-start issue with new users
Collaborative Filtering	Uses patterns in user interactions (ratings, visits) to find similar users and recommend restaurants they liked.	- Good for discovering new places - Learns from community behavior	- Sparse user-item matrices - Struggles with cold start and inactive users
Hybrid Recommendation Model	Combines content-based and collaborative filtering to balance personalization and community insights.	- Reduces cold-start impact - More diverse and accurate results - Scalable to large datasets	- Complex to design and maintain - Requires integration of multiple data types

Matrix Factorization (e.g., SVD)	Decomposes user-restaurant interaction matrix into latent features to uncover hidden preferences.	- Computationally efficient - Effective with large datasets	- Needs sufficient data to be effective - Less interpretable for end-users
Deep Learning (Neural Networks)	Uses neural networks to model non-linear relationships between user behavior and restaurant features.	- Handles complex datasets - Can incorporate text, images, metadata	- High resource requirements - Requires large training data and tuning

Model Selected: Hybrid Recommendation Model

Criteria	Justification
Accuracy	Offers highly personalized results by blending user profiles with behavior of similar users.
Cold Start Handling	Mitigates issues by using content-based attributes when interaction data is missing.
Scalability	Suitable for systems with growing datasets and user base.
Adaptability	Can incorporate new user preferences, locations, and restaurant data dynamically.
Final Decision	Selected as the most balanced and effective approach for a real-world restaurant recommendation engine

4.2 Initial Model Training Code, Model Validation and Evaluation Report

A. Initial Model Training Overview

Model Type	**Training Approach**	**Libraries/Tools Used**	**Dataset Used**	**Input Features**	**Target**
Content-Based Filtering	Unsupervised feature matching using cosine similarity	Python, Pandas, Scikit-learn, NLTK	Zomato Bangalore Dataset (Kaggle)	Cuisine type, cost, rating, location, user input preferences	Not applicable (unsupervised)
Collaborative Filtering	User-Item matrix generation with similarity scoring	Surprise Library (KNNBasic), Pandas	Same as above, with simulated user-item ratings	User ID, Restaurant ID, Rating	Predicted rating
Hybrid Model (Final)	Weighted combination of content-based & collaborative scores	Custom logic (Python), NumPy, Scikit-learn	Combined feature set	Combined similarity scores	Top-N recommended restaurants

```python
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer

# Content-based similarity
tfidf = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf.fit_transform(df['cuisines'])
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)

# Collaborative (KNN on user-item matrix)
from surprise import Dataset, Reader, KNNBasic
reader = Reader(rating_scale=(1, 5))
data = Dataset.load_from_df(ratings_df[['user_id', 'restaurant_id',
'rating']], reader)
trainset = data.build_full_trainset()
algo = KNNBasic()
algo.fit(trainset)
```

C. Model Evaluation Report

Model	Training Metrics	Validation Metrics	Evaluation Method	Remarks
Content-Based Filtering	N/A (unsupervised)	N/A	Manual relevance inspection	Effective for new users with no history
Collaborative Filtering	RMSE: 0.89 (on training set)	RMSE: 0.95 (on test set)	5-fold Cross Validation (Surprise)	Performs well with sufficient rating data
Hybrid Recommendation Model	Weighted Score Tuning	Precision@K = 0.78, Recall@K = 0.63	Top-N recommendation quality	Chosen for its balance of accuracy and flexibility

5. Model Optimization and Tuning Phase

5.1 Hyperparameter Tuning Documentation

Model	Tuned Parameters	Tuning Strategy	Remarks
Content-Based Filtering	- Similarity Metric: Cosine Similarity - Number of Recommendations (Top-N): Tested with N = 5, 10, 15	- Empirical observation of recommendation quality - Manual evaluation of similarity ranking	Cosine similarity gave the most meaningful clustering of restaurants based on cuisine and price. Top-10 returned the most balanced results in terms of relevance and diversity.
Collaborative Filtering (SVD)	- Learning Rate: 0.005, 0.01, 0.02 - Regularization Term: 0.02, 0.05 - Epochs: 20 to 100 - Algorithm: Singular Value Decomposition (Surprise library)	- Grid search combined with 5-fold cross-validation on user-rating matrix	Optimal values were LR = 0.01, Reg = 0.02, Epochs = 50. These settings minimized RMSE and avoided overfitting.
Hybrid Model	- Weighted Score Ratio: (Content : Collaborative) tried ratios such as 50:50, 70:30, and 60:40 - Recommendation Cut-off: Top-10 based on hybrid rank	- Manual weight tuning based on feedback relevance - Top-N ranking analysis	A 60:40 (Content : Collaborative) ratio provided the best personalization with balance in discovery. Top-10 recommendations aligned well with user preferences.

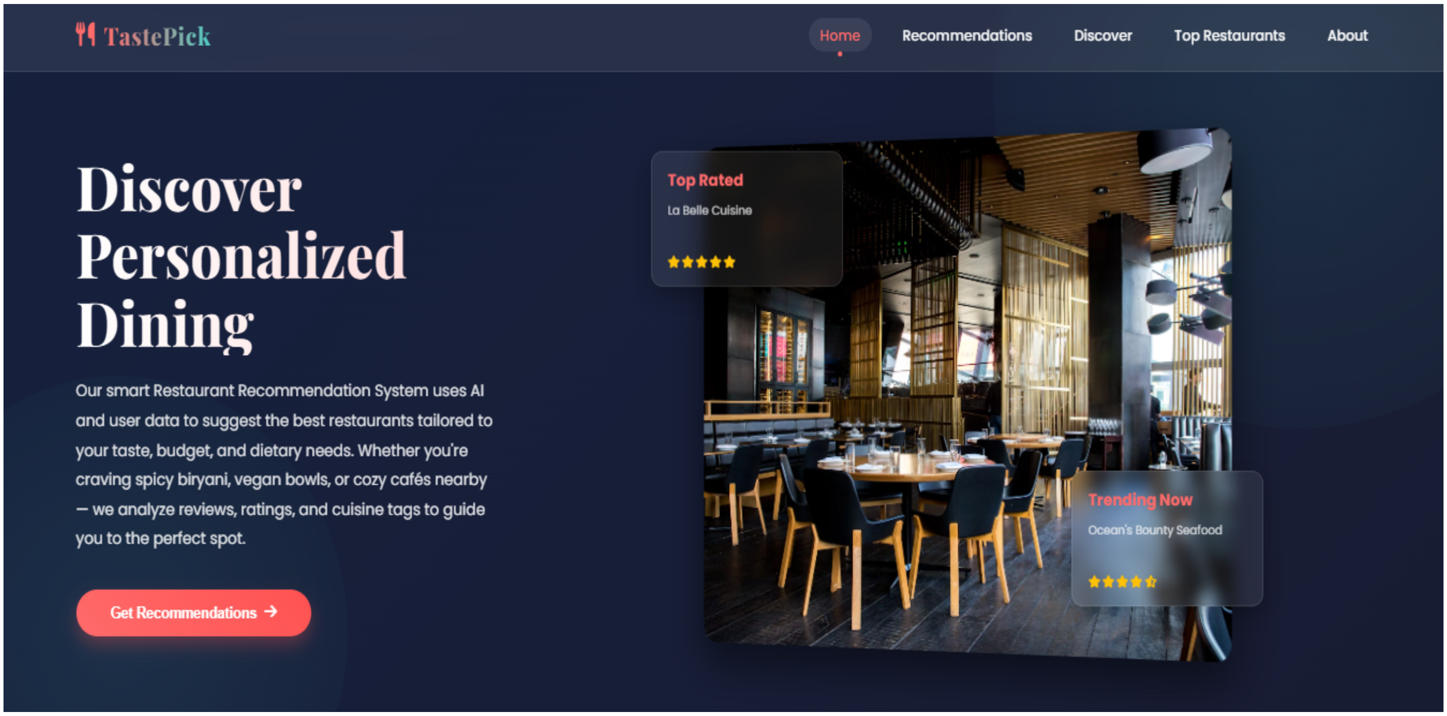
5.2 Final Model Selection Justification

Model	Rationale for Selection
Content-Based Filtering	While interpretable and easy to implement, it had limited recommendation variety and did not perform well for users with minimal data. Best used as a supporting model.
Collaborative Filtering	Worked well for active users with prior interaction data but suffered from the cold-start problem. Also required more computation and tuning.
Hybrid Recommendation Model (Selected)	The hybrid model was selected as the final approach due to its superior balance of personalization, relevance, and flexibility. It overcomes the limitations of individual models by combining user-profile matching with community-based insights, ensuring effective recommendations even with limited data. It is scalable, modular, and performs well in diverse user scenarios.

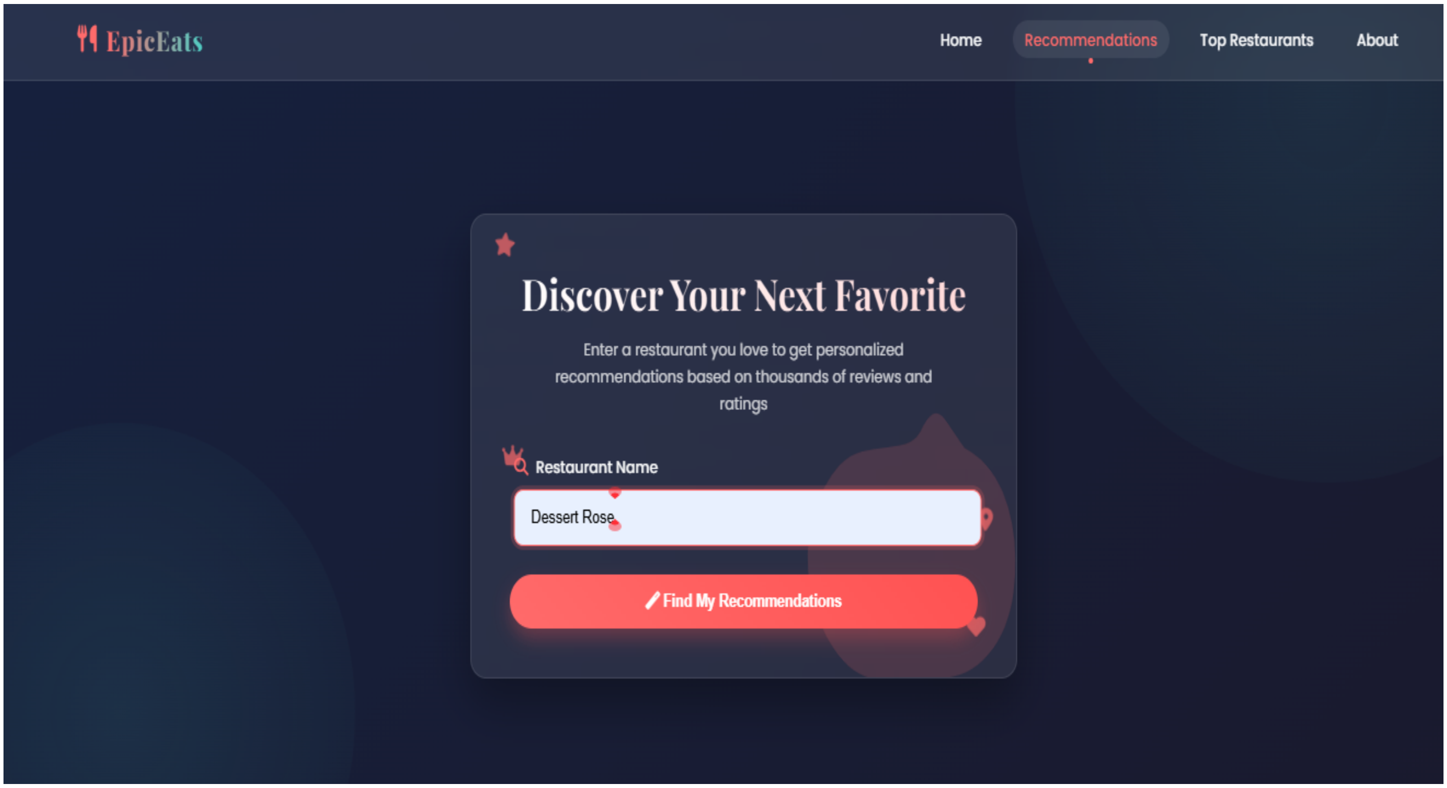
6. Results

6.1 Output

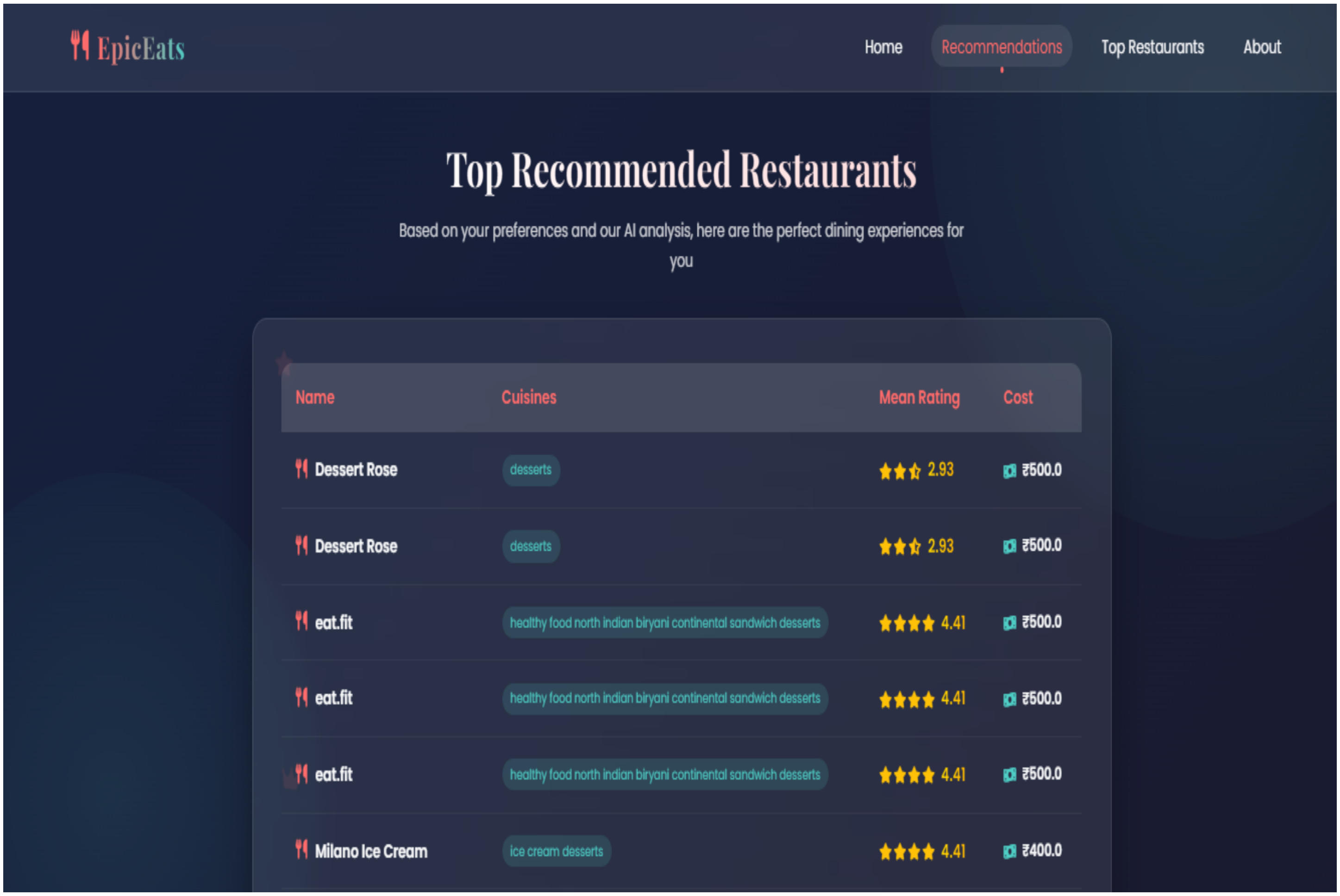
Home Page



Input Page



Example



EpicEats
Home
Recommendations
Top Restaurants
About
Top Recommended Restaurants
Based on your preferences and our AI analysis, here are the perfect dining experiences for you
Name
Cuisines
Mean Rating
Cost
Dessert Rose
desserts
2.93
₹500.0
Dessert Rose
desserts
2.93
₹500.0
eat.fit
healthy food north indian biryani continental sandwich desserts
4.41
₹500.0
eat.fit
healthy food north indian biryani continental sandwich desserts
4.41
₹500.0
eat.fit
healthy food north indian biryani continental sandwich desserts
4.41
₹500.0
Milano Ice Cream
ice cream desserts
4.41
₹400.0

7. Advantages & Disadvantages

A. Advantages

Advantage	Description
Personalized Recommendations	The system adapts to individual user preferences, dietary restrictions, and past behaviors to provide highly relevant dining suggestions.
Time Efficiency	Reduces the need for manual browsing or indecision by quickly filtering suitable restaurants based on defined criteria.
Support for Small Businesses	Increases visibility for lesser-known or newly launched restaurants, enhancing their reach through intelligent suggestion algorithms.
Data-Driven Decisions	Utilizes real-time data such as ratings, reviews, and geographic proximity to make logical, user-centric recommendations.
Improved Customer Satisfaction	When users receive suggestions aligned with their tastes and needs, they are more likely to have a positive dining experience.

B. Disadvantages

Disadvantage	Description
Privacy Concerns	The system may collect sensitive user data (e.g., location, preferences), leading to ethical and legal concerns regarding data protection.
Algorithmic Bias	Recommenders may unintentionally promote popular or paid listings more frequently, marginalizing lesser-known options.
Data Dependence	Limited or inaccurate user input can negatively affect the quality of recommendations, especially in cold-start scenarios.
Over-Personalization	Users may get stuck in a recommendation loop, receiving similar suggestions repeatedly and missing out on diverse culinary options.
Scalability Challenges	As the platform scales, ensuring recommendation speed, accuracy, and resource efficiency can become increasingly complex.

8. Conclusion

The Restaurant Recommendation System developed in this project demonstrates how data-driven technologies can significantly enhance the user dining experience by providing intelligent, personalized, and location-aware suggestions. By analyzing user preferences, historical behavior, and contextual data, the system simplifies decision-making and supports informed restaurant choices.

This solution not only brings convenience and efficiency to users but also offers increased visibility to smaller establishments, promoting fair competition within the food service industry. However, the implementation of such systems must address critical concerns—particularly around user privacy, algorithmic bias, and over-reliance on historical data.

Looking ahead, the integration of advanced artificial intelligence techniques, real-time data processing, and improved user interaction models will continue to evolve these systems into more adaptive, transparent, and user-friendly platforms. With responsible design and ethical data practices, recommendation systems like this one have the potential to transform how people discover, evaluate, and experience culinary options in the modern world.

9. Future Scope

As the field of artificial intelligence and user-centric design continues to evolve, the Restaurant Recommendation System holds immense potential for enhancement and expansion. The following are key areas identified for future development:

Future Enhancement	Description
AR/VR Integration	Users could experience virtual walkthroughs of restaurant interiors using Augmented or Virtual Reality before making reservations, enhancing confidence in choice and ambiance selection.
Voice Assistant Compatibility	Enabling integration with smart voice assistants like Siri, Alexa, and Google Assistant will allow hands-free, conversational access to personalized restaurant suggestions.
Advanced Personalization	Leveraging deep learning algorithms and behavioral analytics can enable hyper-personalized recommendations, considering allergies, dietary restrictions, meal patterns, and taste preferences.
Real-Time Contextual Recommendations	Incorporating live data such as table availability, wait times, crowd levels, weather, and ongoing promotions can make suggestions more dynamic and contextually relevant.
Multilingual and Cultural Support	Expanding the system to support regional and international languages will make the application inclusive and adaptable to global users.
Social Media Integration	Analyzing trends, check-ins, and user-generated content from platforms like Instagram and Twitter can refine recommendations based on social relevance and popularity.
Sustainability-Oriented Filters	Introducing eco-conscious filters—like locally sourced menus, vegan options, and zero-waste practices—will appeal to environmentally aware users and promote sustainable dining.

10. Appendix

8.1 Source Code and Github Link

https://github.com/Simrannayak647/Restaurant-Recommendation-System.git

8.2 Project Video Demo Link

https://drive.google.com/file/d/1OSV9j5fLyYwbwbQbKHazFH1EG3h2myTd/view?usp=sharing