Restaurant Recommendation System

Prepared For
Smart-Internz
Applied Data Science Guided project

By
Digvijay Bajirao Vapilkar
D Y Patil Agriculture and Technical University Talsande

On
13 June 2025

Abstract

This project develops a personalized restaurant recommendation system based on user preferences, location, and dining history. It analyzes factors such as cuisine, price, and ratings to suggest suitable dining options. Machine learning techniques like collaborative and content-based filtering are used for accurate suggestions. The system enhances the dining experience by offering relevant and location-aware recommendations**Final Project Report**

Contents

1.

1.1.

1.1.1. Introduction

1.1.1.1. Project overviews

The **Restaurant Recommendation System** is a smart, data-driven solution designed to help users efficiently discover restaurants that align with their unique preferences and situational contexts. As urbanization and mobile technologies continue to reshape consumer behavior, users are often overwhelmed by the sheer volume of available dining choices across platforms such as Google, Yelp, and Zomato. This leads to **decision fatigue** and suboptimal dining experiences. To solve this, the proposed system leverages a **hybrid recommendation model** combining collaborative filtering, content-based filtering, and geolocation-aware services.The **collaborative filtering component** analyzes historical user behavior, including past restaurant visits, ratings, and interaction patterns, to identify users with similar tastes and recommend restaurants favored by like-minded individuals. Meanwhile, the **content-based filtering module** evaluates restaurant attributes—such as cuisine type, price range, ambiance, and dietary offerings—to match them with explicit user preferences.
To enhance practicality, **geolocation data** is integrated using GPS APIs or IP-based location tracking. This allows the system to dynamically adapt its recommendations based on the user's current position or a specified location, ensuring that results are both **relevant and accessible**. For example, a user seeking budget-friendly vegan food in a new city would receive highly localized and personalized recommendations.

Furthermore, the system is designed with **adaptive learning capabilities**. Using techniques like reinforcement learning or preference feedback loops, the recommendation engine improves over time by understanding user behavior patterns, modifying weightage of features, and incorporating real-time feedback such as likes, bookmarks, or direct reviews.

1.
1.1.
1.1.1.
1.1.1.1. **Objectives**

1. **To design and implement a recommendation engine** that effectively filters and ranks restaurants based on individual user preferences, including food type, cost, ambiance, and dietary needs.
2. **To apply machine learning models**, such as collaborative filtering (user-based and item-based) and content-based filtering, to identify patterns in user behavior and restaurant attributes.
3. **To incorporate location-aware features** using GPS or user-inputted location data, ensuring that recommended restaurants are conveniently accessible to the user.
4. **To gather and analyze restaurant reviews and ratings** from public sources (e.g., Yelp, Google Reviews, or internal datasets) to improve the trustworthiness and relevance of suggestions.
5. **To create a user-friendly interface** that allows users to input preferences,

view recommended restaurants, and interact with the system seamlessly.
6. **To develop a feedback mechanism** that collects user satisfaction data post-visit to refine future recommendations and enhance personalization over time.
7. **To ensure scalability and adaptability** of the system for use in different geographic regions or for integration into existing food delivery or travel applications.

1.

1.1.

1.1.1. Project Initialization and Planning Phase

1.1.1.1. Define Problem Statement

1. Project Initialization and Planning Phase

Problem Statements (Restaurant Recommendation system):

1.

1.1.

1.1.1.

1.1.1.1. Project Proposal (Proposed Solution)

Project Proposal (Proposed Solution)

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel

Resource Requirements

1.

1.1.

1.1.1.

1.1.1.1. Initial Project Planning

Product Backlog, Sprint Schedule, and Estimation

1.

1.1.

1.1.1. Data Collection and Preprocessing Phase

1.1.1.1. Data Collection Plan and Raw Data Sources Identified

Data Collection Plan

Raw Data Sources

1.

1.1.

1.1.1.

1.1.1.1. Data Quality Report

1.

1.1.

1.1.1.

1.1.1.1. Data Preprocessing

Data Preprocessing

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

4.Model Development Phase

1.

1.1.

1.1.1.

1.1.1.1.Model Selection Report

Conclusion:

1.

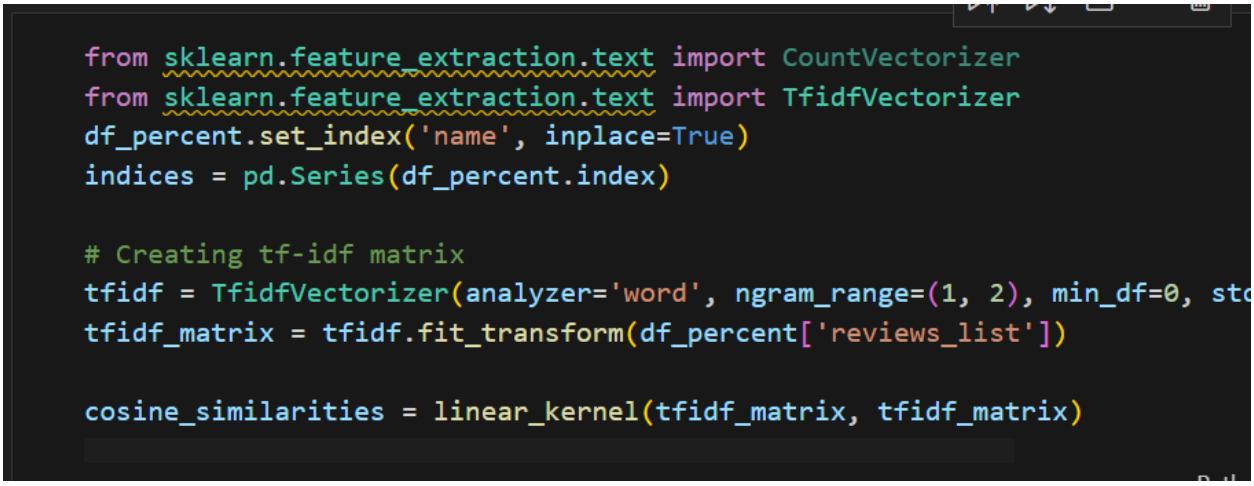
1.1.

1.1.1.

1.1.1.1. Initial Model Training Code, Model Validation and Evaluation Report

Initial Model Training Code, Model Validation and Evaluation Report

Initial Model Training Code (5 marks):



```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
df_percent.set_index('name', inplace=True)
indices = pd.Series(df_percent.index)

# Creating tf-idf matrix
tfidf = TfidfVectorizer(analyzer='word', ngram_range=(1, 2), min_df=0, stc
tfidf_matrix = tfidf.fit_transform(df_percent['reviews_list'])

cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
```

Model Validation and Evaluation Report (5 marks):

1.

1.1.

1.1.1. Model Optimization and Tuning Phase

1.1.1.1. Tunning Documentation

Hyperparameter Tuning

Final Model Selection Justification

Final Model Selection Justification:

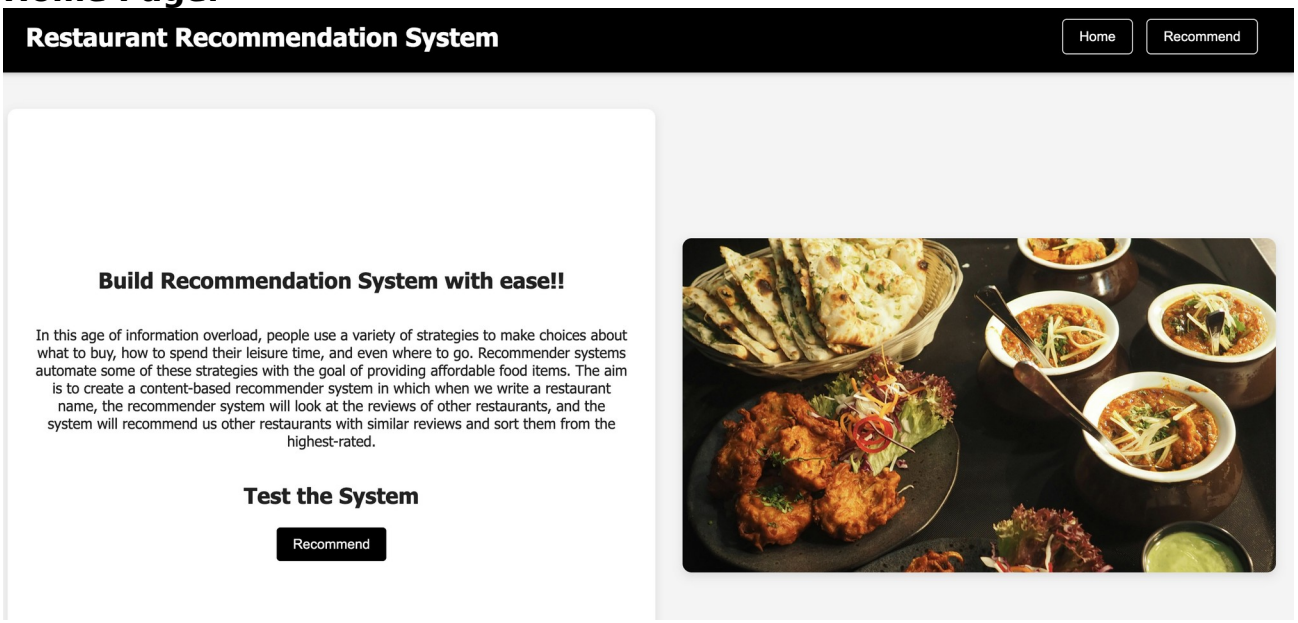
1.

1.1.

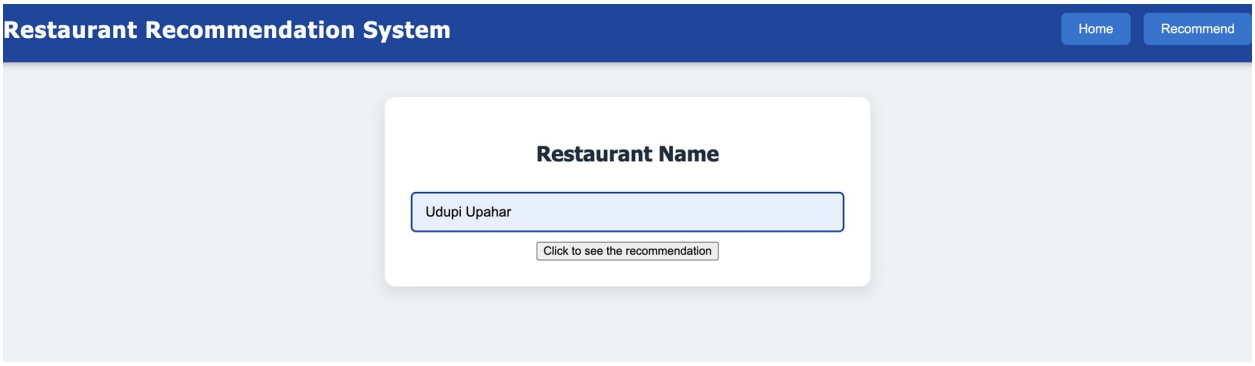
1.1.1. Results

1.1.1.1. Output Screenshots

Home Page:



Input Page: Example :-



Output:

Here are the top recommended restaurants

Name	Cuisines	Mean Rating (out of 5)	Cost (in thousands)
Brahmin's Coffee Bar	south indian	4.85	100.0
Taaza Thindi	south indian	4.7	100.0
Mavalli Tiffin Room (MTR)	south indian	4.41	250.0
Bengaluru Cafe	south indian	4.26	150.0
Sri Laxmi Venkateshwara Coffee Bar	south indian	4.26	100.0
Vidyarthi Bhavan	south indian	4.26	150.0
Prems Graama Bhojanam	south indian	4.11	400.0
South Kitchen	south indian	4.11	100.0
Ayodhya Upachar	south indian north indian chinese street food	4.11	200.0
Ranganna Military Hotel	south indian biryani	3.96	350.0

1.

1.1.

1.1.1. Advantages & Disadvantages

Advantages:

- **Personalized User Experience**: Tailors dining options based on user preferences, dietary needs, and previous behaviour.
- **Time-saving**: Reduces the effort needed to search and choose a restaurant.
- **Improved Discoverability**: Helps smaller or new restaurants gain visibility through recommendations.
- **Data-Driven Decisions**: Uses user ratings, reviews, and location data to make informed suggestions.
- **Enhanced Customer Satisfaction**: Users are more likely to enjoy their meals when recommendations align with their preferences

Disadvantages:

- **Privacy Concerns**: Collecting and analyzing user data (location, preferences) can raise privacy issues.
- **Bias in Recommendations**: Algorithms might favor sponsored listings or high-traffic restaurants, reducing diversity.
- **Dependence on User Data**: Inaccurate or limited data can lead to poor recommendations.
- **Over-Personalization**: Users might be confined to similar choices, missing out on new or diverse dining experiences.
- **Scalability Issues**: Maintaining system accuracy and performance can become challenging as the user base grows.

1.

1.1.

1.1.1. Conclusion

1.

1.1.

1.1.1. **A restaurant recommendation system is a powerful tool for enhancing the dining experience by delivering tailored suggestions based on user behavior, preferences, and location. While it offers significant benefits such as convenience, personalization, and efficient decision-making, it also presents challenges including data privacy, system bias, and the risk of user data dependency. Future advancements in AI, real-time analytics, and user interface technologies promise to make such systems more intelligent, inclusive, and immersive. With careful implementation and ethical considerations, this system can transform how users explore and enjoy culinary optionFuture Scope**

Integration with AR/VR: In the future, users could take virtual tours of restaurants or view their ambiance in AR before booking.

Voice Assistant Compatibility: Integration with Siri, Alexa, or Google Assistant to provide hands-free restaurant suggestions.

Enhanced Personalization: Use deep learning and behavioral analytics to refine suggestions based on dietary restrictions, allergies, and eating habits.

Real-time Data Utilization: Incorporating real-time factors like wait times, special offers, and crowd density for more dynamic recommendations.

Multilingual Support: Expanding the system to support various languages to cater to a global audience.

Social Media Integration: Use of social media trends and check-ins to improve recommendation relevance.

Sustainability Preferences: Factoring in eco-conscious dining choices (e.g., locally sourced, plant-based, or low-waste restaurants).

1.

1.1.

1.1.1. Appendix

1.1.1.1. **Source Code**

[https://github.com/Digu45/DS-Restaurant-System]

1.

1.1.

1.1.1.

1.1.1.1. **Project Video Demo Link :**

Video Demo Link:[https://drive.google.com/file/d/12iJrVa5UiTJv1k-x0lxr2fWat_DrmEZl/view?usp=drive_link]