A Location-Intelligent, Budget-Centric Restaurant Recommender Using Probabilistic Clustering

Ramya B N[1]
[1]Assistant Professor, Artificial Intelligence and Machine Learning, Jyothy Institute of Technology, Bengaluru, Karnataka, India
[1]ramya.bn@jyothyit.ac.in
Phone number - +91 81479 43631 Orcid Id:0000-0001-9312-5176

Diya K Bhat[2]
Artificial Intelligence and Machine Learning, Jyothy Institute of Technology, Bengaluru, Karnataka, India

Vivin Jayanth A M[3]
Artificial Intelligence and Machine Learning, Jyothy Institute of Technology, Bengaluru, Karnataka, India

Samarth S L[4]
Artificial Intelligence and Machine Learning, Jyothy Institute of Technology, Bengaluru, Karnataka, India

Sudarshan D J[5]
Artificial Intelligence and Machine Learning, Jyothy Institute of Technology, Bengaluru, Karnataka, India

***Abstract*— This project presents a scalable and intelligent restaurant recommendation system that integrates Gaussian Mixture Model (GMM) clustering for affordability segmentation with real-time, location-based filtering. By leveraging both automatic GPS detection and manual address input via Google APIs, the system delivers personalized, budget-conscious restaurant suggestions without relying on historical user data. Its key innovations include probabilistic cost-based clustering, dynamic geolocation support, and seamless filtering for relevance and accessibility.**

Keywords— Smart Restaurant Recommender, Gaussian Mixture Model, Machine Learning, Location Detection, Google Maps API Integration, Real-Time Data, Budget Filtering.

1. INTRODUCTION

Choosing a restaurant in cities with many options can be difficult. Most current platforms don't fully consider a person's budget and exact location. This paper introduces a personalized recommendation system that uses machine learning and geolocation to offer tailored suggestions. By leveraging Gaussian Mixture Models for accurate clustering and incorporating real-time data with a user-friendly interface, the system delivers practical, adaptive recommendations for daily use.

2. LITERATURE SURVEY

- **Zeng et al. (2016):**
Proposed a GPS-based hybrid recommender using collaborative filtering; lacked budget filtering and depended on user history. Our system adds budget support and helps first-time users.
- **Sinaga and Yang (2020):**
Discussed choosing K-Means cluster numbers with strong theory but little real-world focus. We use GMM for more flexible, probability-based budget grouping.

- **Veningston and Shanmugalakshmi (2015):**
Used location and time in recommendations but ignored affordability. Our system includes budget limits.
- **Rathnayake (2018):**
Proved Google Maps APIs are effective for real-time location, which we use to find nearby restaurants.
- **Shahapure and Nicholas (2020):**
Emphasized the Silhouette Score for clustering evaluation, supporting our use of it to test GMM results.

3. SCOPE OF RESEARCH

This study presents a restaurant recommendation system that factors in both budget and location. Using Gaussian Mixture Models, it clusters restaurants by affordability and integrates real-time data via external APIs. The system supports GPS-based and manual location input with auto-complete, and works without relying on user history—making it suitable for new and returning users. It is scalable and adaptable, with potential for future improvements like adaptive radius logic and richer clustering using ratings and cuisine types.

4. METHODOLOGY

4.1 Dataset Preparation

The Zomato Bangalore dataset from Kaggle served as the primary data source. Preprocessing steps included removing commas and currency symbols from cost values, converting "approximate cost for two people" into cost per person, and eliminating duplicates and null entries. Column headers were standardized for consistency, resulting in a clean, structured dataset suitable for accurate and efficient clustering.

4.2 Clustering Using Gaussian Mixture Model

This project uses Gaussian Mixture Models (GMM) to cluster restaurants by budget, offering flexible, probabilistic grouping better suited to overlapping price ranges than hard clustering methods like KMeans. The model uses k-means++ for initialization and `covariance_type='full'` to capture detailed pricing patterns. Two clusters were chosen based on the highest silhouette score and practical reasoning—most users view restaurants as either budget-friendly or premium, making the system both accurate and user-friendly.

4.3 Location Services

The system supports two location input methods:

1. **Automatic Detection** – Utilizes browser geolocation to fetch real-time GPS coordinates for the user's current location, as illustrated in figure 3.
2. **Manual Entry** – Enhanced with an autocomplete feature for better user experience, as shown in figure 4, then converted to coordinates using the Google Geocoding API.

4.4 Recommendation System

Restaurants are retrieved using the Google Places API within a specified radius. The results are filtered based on the user's GMM-assigned budget cluster, and duplicates are removed. Final recommendations are sorted by rating, distance, or a combination of both to ensure relevance. Each suggestion includes the restaurant name, address, Google rating, and a link to Google Maps.

4.5 Frontend Interface

The user-friendly web interface lets users enter their budget, search radius, and location (manual or GPS). Clicking “Find Restaurants” triggers clustering and API calls. Results are displayed dynamically with restaurant details and map links.

5. MODELING AND ANALYSIS

This system integrates machine learning with geospatial data to deliver real-time, budget-aware, and location-sensitive restaurant recommendations. The approach combines probabilistic cost clustering using Gaussian Mixture Models (GMM) with geographic filtering powered by the Google Places API. GMM’s soft clustering allows each restaurant to belong to multiple budget segments with varying probabilities, offering flexibility in handling overlapping price ranges and improving recommendation accuracy near category boundaries

.

5.1 Data Modeling

The Zomato dataset was cleaned to retain key fields, with cost converted to "cost per person." A GMM was trained to classify restaurants into low and high-budget clusters, effectively handling overlapping prices. As shown in Figure 2, the model achieved a Silhouette Score of 0.72, indicating well-separated and meaningful clusters.

5.2 Location-Based Filtering

The system supports automatic location detection via device geolocation and manual input geocoded with Google Geocoding API. It then uses the Google Places API to find nearby restaurants within a user-defined radius, ensuring cost-relevant and accessible recommendations.

5.3 Integration and System Logic

The user’s budget is matched to low or high-budget groups via GMM; if it exceeds both, high-end options are shown by default. The system detects location using Google Places API to find nearby restaurants, filters them by budget, removes duplicates, and sorts results by rating or distance. The final list displays each restaurant’s name, rating, address, and a Google Maps link.

5.4 Analytical Insights

Post-deployment analysis showed the two-cluster GMM effectively separated affordable and premium restaurants. Lower-budget users got cafés and fast food, while higher-budget users saw fine dining. GMM’s probabilistic approach handled price overlaps well, and Google geolocation improved speed, context-awareness, and personalization.

6. RESULTS AND DISCUSSION

The restaurant recommendation system was evaluated across multiple test scenarios involving various budget levels, location input methods, and search radii. The outcomes assessed the performance of the clustering model, user interface responsiveness, and real-world relevance of the recommendations.

6.1 Budget-Based Clustering Effectiveness

This project presents a scalable restaurant recommendation system combining real-time location services with budget-aware filtering using GMM. The model achieved a Silhouette Score of 0.72 (Figure 2), indicating well-separated clusters of affordable and premium restaurants. Location is detected via GPS or manual input using Google APIs, and results are filtered by budget and distance. For example, a user with a ₹200 budget in Jayanagar sees cafés or fast food, while a ₹10000 budget shows fine dining options as shown in figure 6 and 7 respectively. In low-density areas, filtering relaxes to improve results. Recommendations include name, rating, address, and a direct map link. The system is reliable and can be enhanced with features like food preferences, adaptive radius, and additional clustering inputs.

6.2 Manual vs Automatic Location Input

The application supports both manual location input and automatic detection via browser geolocation. Manual tests with areas like "Indiranagar" and "Whitefield" confirmed that the Google Geocoding API accurately converts addresses into coordinates. The automatic location feature, using browser-based GPS, provided real-time latitude and longitude, enhancing convenience, particularly for mobile users on the go.

6.3 Search Radius and Location Filtering

Users could define a search radius, controlling the distance over which recommendations were fetched. In urban centers like Koramangala, even a 2 km radius returned sufficient options. In suburban areas, increasing the radius helped populate more results. This filtering, combined with GMM clustering, refined the recommendations both financially and geographically.

6.4 User Interface and Output Quality

The Flask-based web interface was simple yet effective. Inputs such as budget, location, and radius were processed seamlessly, and results were shown in a structured tabular format. Each entry included the restaurant’s name, rating, address, and a link to Google Maps. Duplicate entries were automatically filtered, improving readability and user experience.

6.5 Edge Cases and System Behaviour

To evaluate robustness, several edge cases were tested. Extremely low budgets, such as ₹49, resulted in "Error" messages, as shown in figure 5, ensuring transparency. Invalid locations, like misspelled names, were handled gracefully with prompts for users to retry. In sparse areas or during off hours, fewer results were returned by the Google API, but the system managed these scenarios smoothly without errors.

6.6 Overall Performance

On average, the system responded within seconds for most queries, including live API calls. The combination of GMM clustering and Google's geolocation services allowed the backend to produce relevant, efficient, and highly personalized recommendations. The platform effectively reduced user decision fatigue, ensured relevance through dynamic filtering, and scaled well under varying usage patterns.

7. CONCLUSION

This project introduces a scalable restaurant recommendation system that combines real-time location services with budget-aware filtering using GMM to group restaurants into low and high-budget categories. With a silhouette score of 0.72, the model shows good separation. Google Geocoding and Places APIs provide GPS or manual location-based suggestions, filtered by budget and distance. Results include name, rating, address, and map link. The system handles low budgets and sparse areas well, offering fast, relevant results. Future improvements could include food preferences, adaptive search radius, and additional clustering data like ratings and cuisine types.

8. REFERENCES

[1] H. Poddar, "Zomato Bangalore Restaurants Dataset," *Kaggle*. [Online]. Available: https://www.kaggle.com/datasets/himanshupoddar/zomato-bangalore-restaurants

[2] Google Developers, "Places API Documentation," *Google Maps Platform*. [Online]. Available: https://developers.google.com/maps/documentation/places/web-service/overview

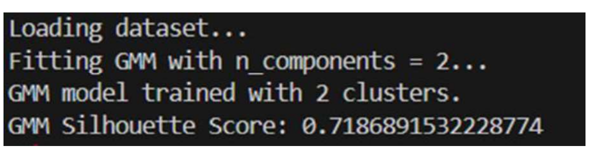
[3] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson Education, 2006.

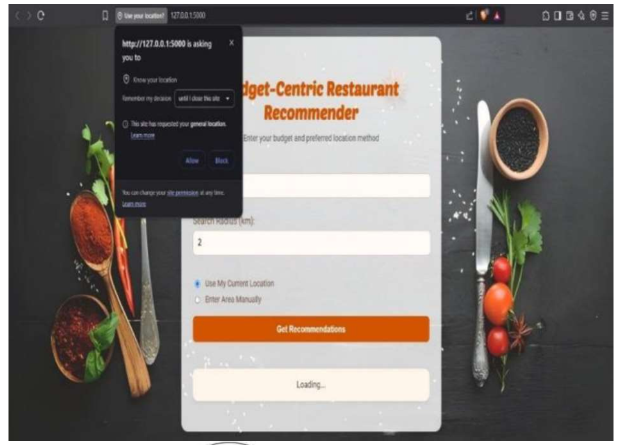
[5] Flask Project, "Flask Documentation." [Online]. Available: https://flask.palletsprojects.com/

[6] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

9. FIGURES



```
Loading dataset...
Fitting GMM with n_components = 2...
GMM model trained with 2 clusters.
GMM Silhouette Score: 0.7186891532228774
```



fi

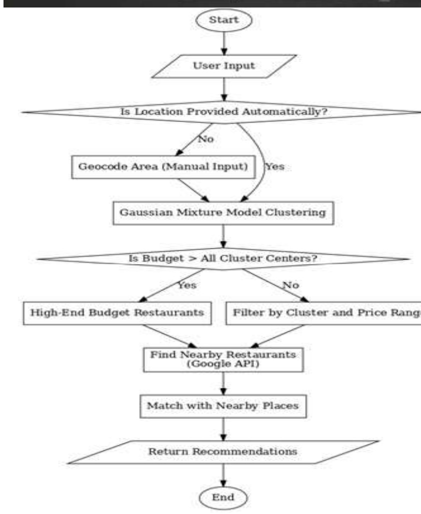


Fig. 1. Smart Restaurant Recommendation System Workflow Diagram

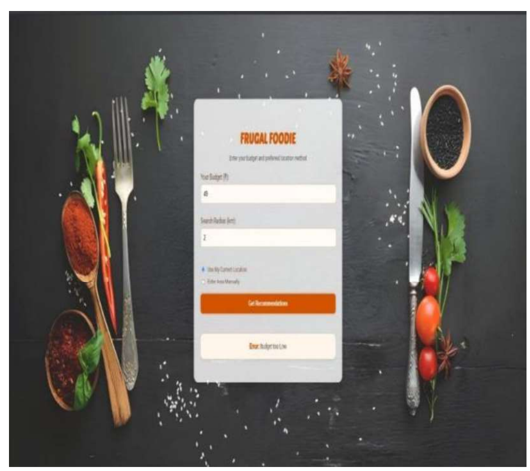


Fig. 3. Auto-detected user location

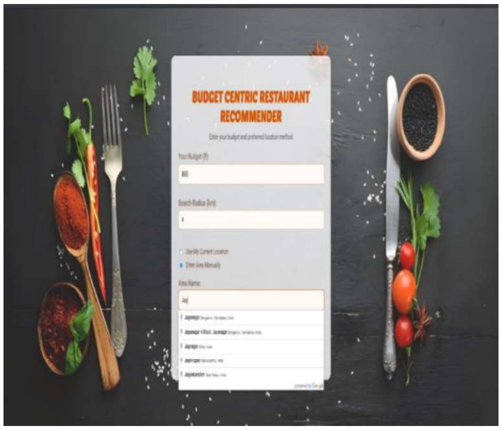


Fig. 4. Manual location input

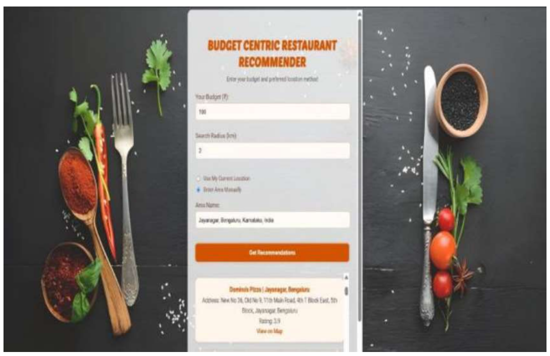


Fig. 5. Example of “Error” message for very low-budget input.

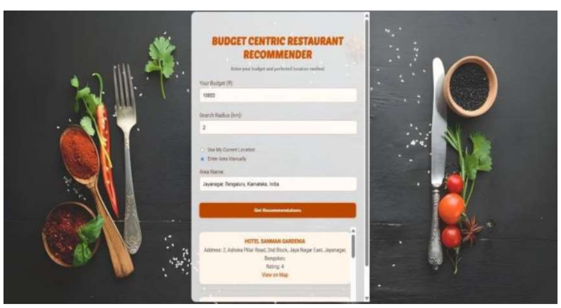


Fig. 6. Low budget output

Fig. 7. High budget output