Restaurant Recommender System using Unsupervised Learning

Anushka Sagar[1]

[1]

[2]Department of Computer science and engineering, Lovely Professional University , Punjab, India

ABSTRACT This abstract presents a comparative analysis of unsupervised methods K-means, DB-scan, K-medoids for Restaurant Recommendation, that suggests restaurants to users based on their location, cuisine preferences, and reviews. using a dataset collected from ChatGPT. The study explores preprocessing steps, feature engineering techniques using pipeline, and model training strategies customized for best algorithm. By evaluating Silhouette Score for all the methods.

INDEX TERMS Clustering algorithms, K-means clustering, DBSCAN, K-medoids, Elbow method, Feature Engineering, Unsupervised learning, Evaluation Matrices

INTRODUCTION

Recommendations have become an integral part of today's decision-making process by providing personalized recommendations based on user preferences and contextual information. In this work, we present a novel clustering-based approach to improve the efficiency and accuracy of restaurant recommendation systems. Our system groups restaurants based on their attributes such as location, food, rating, overall review, and average price per person using unsupervised learning techniques such as K-Means, DBSCAN, and KMedoids. The goal of this project is twofold: First, to compare the results of different methods using indicators such as Silhouette Score; Second, to generate recommendations for users based on specific users (such as Cities, Measurements, and Budgets), K-Means emerged as the best common process for our data. To demonstrate excellent performance and additional computing power, we implemented an interactive interface that allows users to access their preferred design restaurant recommendation list.

METHODOLOGY

A Data Collection:

The dataset used in this Restaurant recommender System project was sourced from ChatGPT. The dataset comprises 'Restaurant-ID', 'Name', 'Location', 'City', 'Cuisine', 'Rating', 'Price Range', 'Total Reviews', 'Average Cost for Two (INR) collected from google and ChatGPT.

B Data Preprocessing:

• Feature Selection and Cleaning: Selected essential features such as city, cuisine type, and cost for two, ensuring the dataset is free from null or erroneous values. • Feature Scaling and Encoding: Used a pipeline to preprocess the data. Numerical features (e.g., rating, total reviews) were scaled using StandardScaler, while categorical features (e.g., city, location) were one-hot encoded.

C Hyperparameter Tuning:

Each clustering algorithm underwent hyperparameter tuning:

- K-Means: Optimal k was determined by testing values ranging from 2 to 10 and evaluating Silhouette Scores.
- DBSCAN: Various combinations of eps and min-samples were tested to find the best-performing parameters.
- K-Medoids: The optimal k value was identified similarly to K-Means.

D Clustering Algorithms:

- K-Means: Implemented K-Means for clustering based on Euclidean distances, optimizing the number of clusters using the Elbow Method.
- DBSCAN: Explored density-based clustering to group restaurants without prior knowledge of cluster count, focusing on spatial density.
- K-Medoids: Used K-Medoids to minimize the sum of dissimilarities within clusters, making it robust to outliers. Each algorithm was evaluated using Silhouette Score to quantify cluster quality and determine the most effective clustering approach.

E Recommendation System Development:

- Dynamic Filtering: Designed a filtering system to recommend restaurants based on user-specified constraints such as city, minimum rating, and maximum budget.

• Cluster-based Recommendations: Mapped restaurants to their respective clusters, ensuring recommendations are both relevant and diverse.

F Evaluation and Visualization:

• Clustering Performance Comparison: Silhouette Scores of all algorithms were plotted for easy comparison.
Additionally, individual cluster visualizations were created for K-Means, DBSCAN, and K-Medoids. • User Validation: Conducted a simulated user study to assess the relevance and quality of recommendations.

RESULTS

A Dataset:

The dataset used in this Restaurant recommender System project was sourced from ChatGPT. The dataset comprises 'Restaurant-ID', 'Name', 'Location', 'City', 'Cuisine', 'Rating', 'Price Range', 'Total Reviews', 'Average Cost for Two (INR) collected from google and ChatGPT.

B Evaluation Matrices:

Evaluating each algorithm used for recommender system using silhouette score which represents that it ranges from -1 to 1:

Silhouette Score = 1: Indicates that the points are well clustered, meaning the data points are well-matched to their own cluster and far from other clusters.
Silhouette Score = 0: Indicates that the points are on or very close to the decision boundary between two clusters. This suggests that the clusters are not very distinct. Silhouette Score ¡ 0: Indicates that the points might have been incorrectly assigned to the wrong clusters.

$$s = \frac{b - a}{\max(a, b)}$$

FIGURE 1. Silhouette score : a(i) is the average distance between point iii and all other points in the same cluster (cohesion). b(i) is the minimum average distance from point iii to all points in any other cluster (separation).

Clustering Algorithm	Silhouette Score	cluster	DBSCAN Parameters
K-Means	0.3746	3	N/A
DBSCAN	0.3566	N/A	(eps=0.3, min = 5)
K-Medoids	0.3669	6	N/A

FIGURE 2. Comparison table: Comparison table of score between all three of the algorithm , hence we can conclude the K-means is the best. .

C. Visual representation :

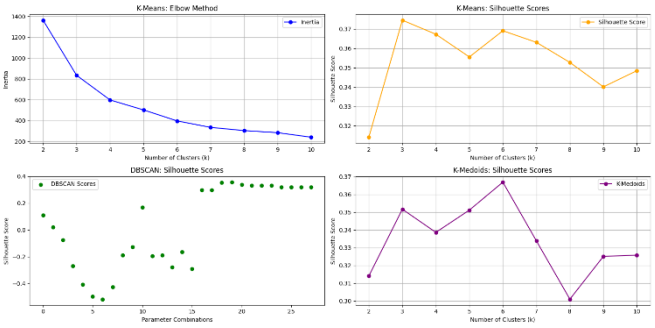

FIGURE 3. Elbow method and Silhouette score "Fig.3" the elbow method for optimal number of cluster in k-mean and also silhouette score graph for each algorithm.

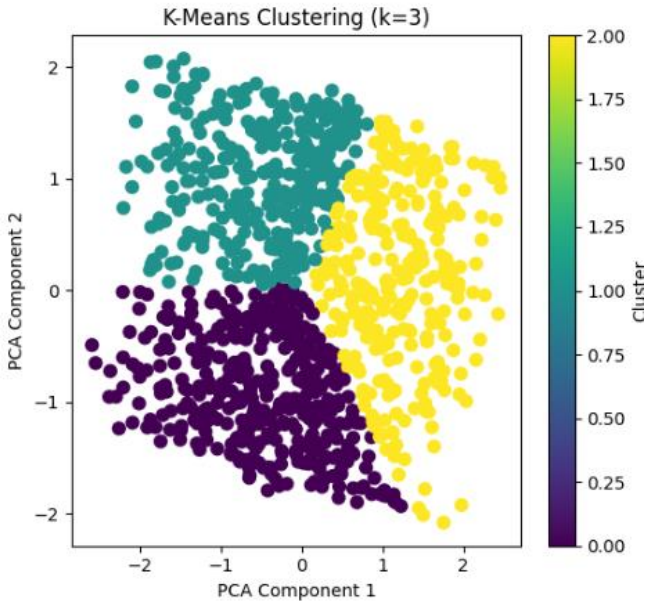

FIGURE 4. K-means Clustering"Fig.4" the graph shows K-means cluster where the data is divided into three different cluster by calculating distance.

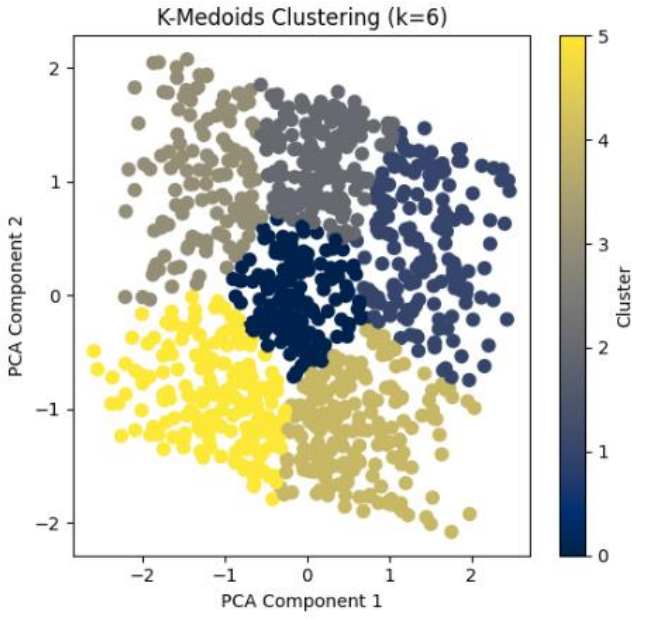

FIGURE 5. K-medoids clusters "Fig.5" The following graph shows k-medoids clustering technique by diving data into 6 cluster and no noise is captured in k-means clustering

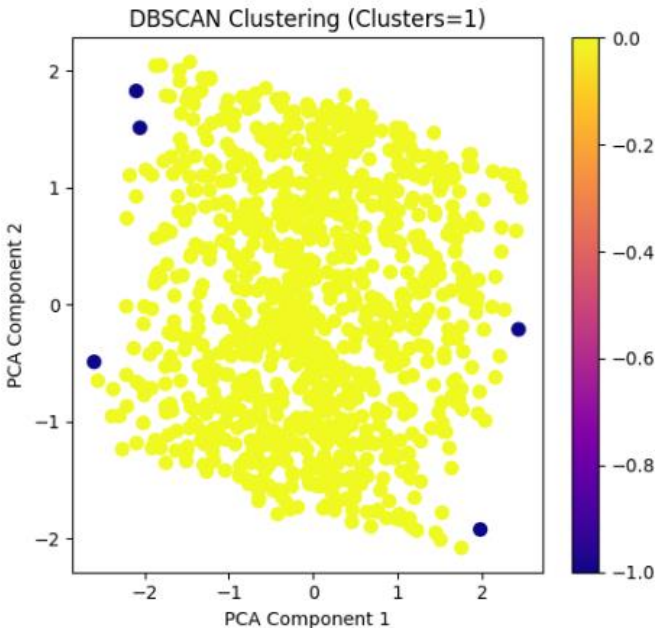

FIGURE 4. DBScan Clustering "Fig.6" the graph shows DBScan cluster where the data is clustered into one single cluster and rest as noise.

I. CONCLUSION

In this study, we investigated different methods, focusing on K-Means, DBSCAN and K-Medoids and their ability to cluster data well. The performance of each algorithm was analysed in different scenarios using the Silhouette Score, a measure of cluster performance. Silhouette scored the highest at 0.3746, making it the first choice for applications with a predefined set of categories. Capable of handling noisy and irregular clusters, K-medoids demonstrates its strengths on many complex files, showing an average performance of 0.3669 Silhouette score. However, in this case, DBScan scored the lowest Silhouette score (0.3556), indicating that it may not be suitable for this data but may still be useful in the environment, especially for those who need representative clusters. In summary, this research provides insight into the strengths and weaknesses of cluster analysis algorithms. While K-means is the most useful for this task, the choice of integration depends on the nature of the product and the specific problem to be solved. Future work could investigate the application of this algorithm to different datasets, including additional performance metrics, and examine the implications of modifying the algorithm to improve results.

SOURCE CODE AND SCRIPTS

Following Source code and results of my work with dataset used. https://github.com/051821/Restaurantrecommender.

LITRATURE REVIEW

[1]This paper provides a comprehensive review of clustering algorithms, including K-Means and its limitations, and explores alternative methods such as DBSCAN and K-Medoids.

[2]This paper introduces DBSCAN, one of the key algorithms evaluated in your research, and presents its advantages in identifying clusters of varying shapes and densities, as well as handling noise.

[3]This book provides an in-depth exploration of various clustering techniques, including K-Medoids, and discusses methods for evaluating the quality of clustering results.

REFERENCES

Jain, A. K. (2010). Data Clustering: 50 Years Beyond K-Means. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(7), 1447-1464. DOI: 10.1109/TPAMI.2008.239[1]

Ester, M., Kriegel, H. P., Sander, J., Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in
Large Spatial Databases with Noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), 226-231.[2]

Kaufman, L., Rousseeuw, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley- Interscience.[3]