Swiggy Restaurant Recommendation System using K Means Clustering

Name – Rahul Raj

Batch - DS-C-WE-E-B89

1. Introduction

In recent years, online food delivery platforms such as Swiggy have witnessed significant growth due to rapid urbanization, busy lifestyles, and increasing dependence on digital services. With the expansion of these platforms, users are provided access to thousands of restaurants offering a wide variety of cuisines, price ranges, and dining experiences. While this abundance of choices offers flexibility, it also introduces a major challenge in decision-making. Users often struggle to identify restaurants that best align with their preferences, such as location, cuisine type, budget, and quality. This challenge highlights the growing importance of intelligent recommendation systems that can simplify the selection process by providing personalized suggestions. Recommendation systems have become a core component of modern digital platforms, playing a vital role in enhancing user engagement and satisfaction. By analysing historical data and user preferences, these systems reduce information overload and help users make faster and more informed decisions. In the context of food delivery platforms, effective recommendation systems not only improve customer experience but also increase platform retention and support restaurant visibility. As a result, recommendation engines have become an essential tool in the food technology and data analytics domain. This project focuses on the development of a Restaurant Recommendation System inspired by Swiggy, using Python-based data analytics and machine learning techniques. The system recommends restaurants based on user-selected preferences such as city, cuisine, rating, and cost. Instead of relying on manual browsing, users can interact with an intelligent system that processes their inputs and suggests relevant restaurants. To ensure accessibility and ease of use, the system is implemented as an interactive web application using stream lit, allowing real-time input and

output visualization. The project follows a structured and systematic workflow. Initially, raw restaurant data is collected and cleaned to remove inconsistencies, duplicates, and missing values. This is followed by data pre-processing, where categorical features such as city and cuisine are transformed into numerical format using encoding techniques. These steps ensure that the dataset is suitable for machine learning algorithms. The core recommendation logic is implemented using K-Means clustering, an unsupervised learning algorithm that groups restaurants with similar characteristics. Based on the cluster that best matches the user's preferences, the system generates appropriate restaurant recommendations. The primary objective of this project is to demonstrate how real-world business data can be effectively analysed and converted into actionable recommendations. The system not only assists users in discovering suitable restaurants but also showcases how data-driven approaches can support business decision-making. Additionally, this project serves as a practical example of integrating machine learning models with web-based dashboards, bridging the gap between analytics and user interaction. Overall, the project highlights the practical application of data analytics, machine learning, and visualization techniques in building intelligent, scalable, and user-centric recommendation systems.

2. Tools Used

Python

Python is the primary programming language used to develop the entire restaurant recommendation system. It is widely adopted in data analytics and machine learning due to its simplicity, readability, and vast ecosystem of libraries. In this project, Python is used for loading and processing the restaurant dataset, performing data cleaning operations such as handling missing values and removing duplicates, and implementing the recommendation logic. Python also acts as the backbone that integrates data pre-processing, model building, and application development into a single workflow, ensuring smooth execution and reproducibility of results.

K-Means Clustering Model

K-Means is an unsupervised machine learning algorithm used to identify patterns and similarities within the restaurant data. In this project, K-Means clustering groups restaurants into clusters based on encoded numerical features. Restaurants within the same cluster share similar characteristics, such as location, cuisine type, price range, and rating. When a user provides preferences through the dashboard, the system identifies the cluster to which the input belongs and recommends restaurants from that cluster. This approach improves recommendation relevance without requiring labelled output data.

Scikit-learn

Scikit-learn is a powerful machine learning library in Python used for data pre-processing and model implementation. In this project, Scikit-learn is used to apply **One-Hot Encoding** on categorical features such as city and cuisine, converting them into numerical format suitable for machine learning algorithms. It is also used to implement the **K-Means clustering algorithm**, which group's restaurants based on similarity in features like rating, cost, city, and

cuisine. Additionally, Scikit-learn provides efficient tools for model fitting, distance computation, and consistent transformation of both dataset and user input.

Stream lit

Stream lit is used to build an interactive and user-friendly web dashboard for the recommendation system. It allows users to input preferences such as city, cuisine, rating, and cost through dropdowns and sliders. Stream lit dynamically processes these inputs, passes them to the recommendation engine, and displays the recommended restaurant names in real time. Its simplicity enables rapid development of data-driven applications without requiring extensive frontend knowledge, making it ideal for showcasing machine learning projects.

Pickle (Model & Encoder Storage)

Pickle is used to save trained models and encoders for reuse in the Stream lit application. The One-Hot Encoder and K-Means model are serialized and stored as .pkl files. This ensures that the same pre-processing logic and clustering behaviour are applied during prediction as during training. Using Pickle improves efficiency, maintains consistency, and allows the application to load models instantly without retraining.

3. Data Processing and Encoding

Data pre-processing is a crucial step in the development of a reliable and accurate restaurant recommendation system, as the quality of input data directly impacts the performance of machine learning models. In this project, the raw restaurant dataset obtained in CSV format contained both categorical and numerical attributes such as restaurant name, city, cuisine, rating, rating count, and cost. Before applying any machine learning techniques, the dataset was carefully analysed to identify inconsistencies, missing values, and duplicate records. Duplicate rows were removed to avoid bias in clustering and similarity computations, while missing values in important numerical columns such as rating and cost were handled either by removal or appropriate imputation to ensure data consistency and reliability.

Once the data cleaning process was completed, the next step involved selecting relevant features for pre-processing. Since machine learning algorithms require numerical input, categorical variables such as city and cuisine could not be used directly in their original textual form. Therefore, these features were transformed using encoding techniques. Numerical attributes like rating, rating count, and cost were retained as they are, as they already represent meaningful quantitative information. The cleaned dataset was saved separately as cleaned_data.csv to preserve human-readable information such as restaurant names, which are later used for displaying recommendations.

To convert categorical data into a machine-readable format, One-Hot Encoding was applied using the Scikit-learn library. One-Hot Encoding transforms each category into a binary vector, where the presence of a category is represented by a value of 1 and absence by 0. In this project, encoding was applied specifically to the city and cuisine columns, as they play a major role in determining restaurant similarity. This approach ensures that no unintended ordinal relationships are introduced between categorical values, which could otherwise distort clustering results.

After encoding, all features in the dataset became numerical, making them suitable for distance-based algorithms such as K-Means clustering and cosine similarity. The encoded dataset was stored as encoded_data.csv, which contains only numerical features required for computation and excludes non-essential display columns like restaurant name. A key consideration during pre-processing was maintaining index alignment between cleaned_data.csv and encoded_data.csv. This alignment ensures that recommendations generated using encoded data can be accurately mapped back to their original restaurant details.

Additionally, the fitted one-Hot Encoder was saved as a serialized file (encoder.pkl) using Pickle. This step is essential for maintaining consistency between training and prediction phases. During real-time user interaction in the Stream lit application, the same encoder is reused to transform user inputs into the same feature space as the training data, preventing data leakage and encoding mismatch.

Overall, the data pre-processing and encoding stage ensures clean, structured, and machine-readable input for the recommendation engine. By carefully handling missing values, encoding categorical features, and maintaining proper data alignment, the project establishes a strong foundation for accurate restaurant clustering and personalized recommendations.

4. Recommendation Methodology

The restaurant recommendation system in this project is built using a clustering-based approach, specifically the K-Means algorithm, which is an unsupervised machine learning technique. Unlike traditional supervised models that require labelled outputs, K-Means identifies hidden patterns in data by grouping similar data points into clusters based on feature similarity. This makes it well suited for recommendation systems where explicit user-restaurant preference labels are unavailable. The core objective of using K-Means in this project is to group restaurants with similar characteristics such as city, cuisine type, price range, and rating, and then recommend restaurants from the most relevant cluster based on user preferences. The recommendation methodology begins with the use of a fully numerical, encoded dataset generated during the pre-processing stage. Categorical attributes such as city and cuisine are converted into numerical form using One-Hot Encoding, while numerical features such as rating, rating count, and cost are retained. These features collectively represent each restaurant as a point in a multi-dimensional feature space. Since K-Means relies on distance calculations to form clusters, this uniform numerical representation ensures accurate similarity measurement between restaurants. K-Means clustering works by dividing the dataset into a predefined number of clusters, denoted by K. The algorithm initializes K centroids and iteratively assigns each restaurant to the nearest centroid based on Euclidean distance. After assignment, centroids are recalculated as the mean position of all points within a cluster. This process continues until cluster assignments stabilize or convergence is achieved. In this project, the value of K is selected using techniques such as the elbow method to balance cluster compactness and interpretability. Once the clustering model is trained on the encoded dataset, each restaurant is assigned a cluster label. These cluster labels represent groups of restaurants that share similar attributes. The trained K-Means model is then saved using Pickle so that it can be reused during real-time recommendations without retraining. When a user provides preferences such as city, cuisine, budget, and expected rating through the Stream lit interface, these inputs are encoded using the same One-Hot Encoder and transformed into the same feature space as the training data. The encoded user input is passed to the trained K-Means model to determine the closest cluster. Restaurants belonging to the same cluster are considered most relevant to the user's preferences. From this cluster, recommendations are generated by filtering restaurants based on additional constraints such as minimum rating or maximum cost. This approach ensures that

recommendations are both similar in characteristics and aligned with user expectations. A critical step in the recommendation pipeline is mapping the results back to the original, human-readable dataset. Since clustering is performed on the encoded dataset, the resulting restaurant indices are mapped back to cleaned_data.csv using index alignment. This allows the system to display meaningful information such as restaurant names, locations, and ratings instead of encoded values. Overall, the K-Means-based recommendation methodology provides an efficient and scalable solution for restaurant recommendations. By grouping similar restaurants and recommending from the most relevant cluster, the system delivers personalized suggestions while maintaining computational simplicity and real-time performance within the Stream lit application.

5. Observation

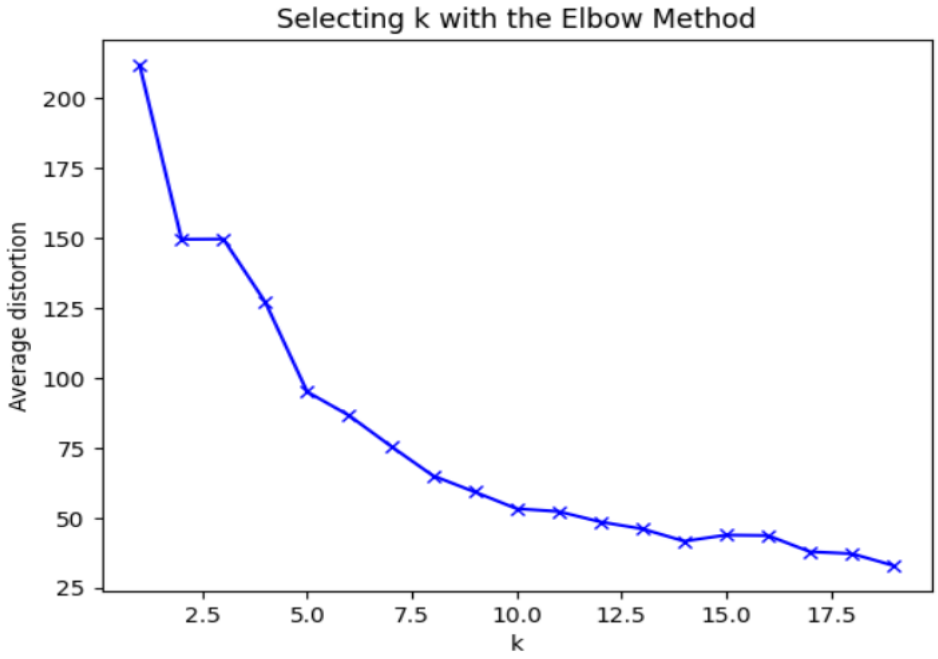

Fig1: Clustering with Elbow Method

[71]:

	name	city	rating	rating_count	cost	cuisine	cluster
49597	Slurpy Shakes	Patna	4.6	50	150	Beverages,Ice Cream	13
49579	Madras Dosa House	Patna	4.3	50	2	South Indian,Chinese	13
49612	Meera hut	Patna	4.3	50	2	North Indian,Fast Food	13
49776	Open with Smile	Patna	4.2	20	150	Chinese,North Indian	13
49596	Sundae Everyday Ice Creams	Patna	4.1	20	100	Ice Cream,Desserts	13
49702	Momo's World Cafe	Patna	4.0	20	100	Chinese,Snacks	13

Fig2: Dashboard of Restaurant Recommendation

6. Future Scope

The future scope of the Swiggy-style restaurant recommendation system is broad and offers significant opportunities for enhancement in terms of intelligence, scalability, and real-world applicability. One major extension of this project would be the integration of user behaviour data such as past orders, browsing history, and search patterns to move from a purely content-based recommendation approach to a hybrid recommendation system. By incorporating collaborative filtering techniques, the system could learn from similar users' preferences and provide more personalized and dynamic recommendations. Additionally, advanced machine learning models such as hierarchical clustering, DBSCAN, or deep learning–based embedding models could be explored to capture more complex relationships between restaurants and user preferences beyond what K-Means clustering can represent. Another important future improvement lies in incorporating real-time data sources. Currently, the system operates on a static CSV dataset; however, integrating live data through APIs would enable real-time updates of restaurant availability, pricing changes, offers, and ratings. This would significantly enhance the relevance and reliability of recommendations. Furthermore, the inclusion of natural language processing techniques could allow users to provide preferences in free-text form, such as "budget-friendly North Indian food in Kolkata," making the system more intuitive and user-centric.

From a business intelligence perspective, the system can be extended to provide analytical dashboards for restaurant owners and administrators. These dashboards could display insights such as popular cuisines in specific cities, pricing trends, customer rating distributions, and cluster-wise performance analysis. Such insights would help businesses optimize menu offerings, pricing strategies, and marketing campaigns. Predictive analytics could also be added to forecast demand trends based on historical data, seasonal patterns, and customer preferences. The stream lit application itself can be further enhanced by improving the user interface and experience. Features such as interactive maps, advanced filters, restaurant images, and user reviews can be added to make the dashboard more engaging and informative. The application could also be deployed on cloud platforms such as AWS, Azure, or He Roku to support large-scale usage and multi-user access. Integration with mobile platforms could further increase accessibility and reach. From a data engineering standpoint, future work could involve replacing CSV-based storage with relational or NoSQL databases to handle larger datasets efficiently. Implementing automated data pipelines and model retraining mechanisms would

ensure that the recommendation system remains up to date as new data is added. Security and privacy considerations can also be strengthened by implementing user authentication, access control, and data encryption mechanisms. Overall, the project provides a strong foundation for a real-world recommendation system, and with the incorporation of advanced machine learning techniques, real-time data integration, enhanced visualization, and scalable deployment, it has the potential to evolve into a fully functional, industry-grade restaurant recommendation platform.

7. Conclusion

This project successfully demonstrates the design and implementation of a restaurant recommendation system inspired by Swiggy, using data analytics and machine learning techniques. The primary objective of the project was to recommend restaurants based on user preferences such as city, cuisine, rating, and cost, and this objective was effectively achieved through systematic data pre-processing, feature encoding, and clustering-based recommendation methodology. By transforming raw restaurant data into a structured and machine-readable format, the project establishes a strong foundation for generating accurate and meaningful recommendations.

The use of One-Hot Encoding enabled effective handling of categorical features such as city and cuisine, ensuring compatibility with machine learning algorithms. The application of the K-Means clustering algorithm allowed restaurants with similar characteristics to be grouped together without the need for labelled training data. This unsupervised approach proved to be efficient and scalable, making it suitable for real-world recommendation scenarios where explicit user feedback may not always be available. Mapping the clustering results back to the cleaned dataset ensured that recommendations remained interpretable and user-friendly.

The integration of Streamlet played a vital role in transforming the analytical model into an interactive and accessible application. The user-friendly dashboard allows users to input their preferences and instantly view relevant restaurant recommendations, bridging the gap between machine learning models and real-world usability. The modular structure of the project, including separate cleaned and encoded datasets along with serialized models, ensures consistency, reproducibility, and ease of future enhancement.

Overall, this project highlights the practical application of machine learning and data analytics in solving real-world business problems. It not only enhances user decision-making by providing personalized restaurant suggestions but also offers valuable insights that can support business strategies and operational planning. The system serves as a solid proof of concept for recommendation engines and lays a strong foundation for future expansion into more advanced, scalable, and intelligent recommendation platforms.