Restaurant Recommendation System

A PROJECT REPORT BY

Name: Tripti Bhardwaj

ABSTRACT

The restaurant industry is undergoing a transformative shift with the advent of digital platforms, making restaurant recommendation systems a crucial tool for aiding users in making informed dining choices. This documentation presents the development of a robust restaurant recommendation system that harnesses machine learning techniques to provide tailored suggestions to users. The system capitalizes on diverse attributes including reviews, ratings, cuisines, and costs to generate personalized restaurant recommendations.

The documentation offers an in-depth exploration of the project's methodology. It delves into data preprocessing steps, including data loading, handling missing values, and data type conversions. Notably, the creation of the "Mean Rating" feature through user reviews and its normalization using MinMaxScaler is elaborated upon.

Furthermore, the intricate process of calculating cosine similarity is expounded upon. The generation of TF-IDF vectors for reviews and their utilization in constructing a cosine similarity matrix is discussed. The documentation encompasses the creation of recommendation functions based on various attributes like cuisines, mean rating, and cost, enabling users to explore restaurants aligned with their preferences.

Through comprehensive experimentation, the project's efficacy in providing relevant and accurate recommendations is evaluated. The documentation concludes with a reflection on the outcomes, highlighting the system's success in aiding users' decision-making processes. Moreover, potential avenues for enhancement, such as incorporating collaborative filtering or sentiment analysis, are identified, paving the way for future developments.

In summation, this documentation unveils a sophisticated restaurant recommendation system that capitalizes on machine learning techniques to empower users with tailored dining recommendations, augmenting the dining experience in the digital age.

1.1 INTRODUCTION

In an era marked by increasingly diverse culinary preferences and the abundance of dining establishments, choosing the right restaurant can be a daunting task for consumers. The overwhelming number of choices, combined with varying factors such as location, cuisine, budget, and personal taste, often leave individuals in search of an efficient and personalized solution. This problem statement outlines the need for the development of a Restaurant Recommendation System (RRS) to address this issue and provide users with tailored restaurant recommendations.

Recommender Systems or Recommendation Systems are simple algorithms that aim to provide the most relevant and accurate items (products, movies, events, articles, food, restaurants) to the user (customers, visitors, app users, readers) by filtering useful stuff from a huge pool of information. Recommendation engines discover data patterns in the data set by learning consumers' choices and produce the outcomes that correlates to their needs and interests. Our project is to develop a similar recommendation system related to restaurants.

Our aim is to build a restaurant recommendation system that provides personalized restaurant recommendations to users. Since different people have different food preferences and dietary restrictions, we perform careful feature selection to take advantage of the information reflected in a user's reviews.

Furthermore, since dining is frequently a communal activity, we generated recommendations not just for an individual user, but also for a group of users. This system needs to consider the information of all individuals in a group and recommend restaurants that satisfy the group of users according to some criterion. Although reviews provide valuable recommendations about restaurants, reading reviews for businesses can be time consuming. People go through several pages of recommendations published on Yelp just wanting to find the right restaurant, which has the most potential to satisfy their personal appetite or their own food preference. Above that, when people visit a new place, move or travel, users have to go through the same search procedure again. This can be tedious and sometimes annoying.

Food is not just a necessity of life. The food we eat represents our culture, tradition, and values. The norms and values of a place can be significantly related to varieties of food available there. Eventually, the highest rated hotel is being recommended to the user by the restaurant recommended system. Our application takes the food preference and ratings into consideration to recommend food to the users. The application uses item based content based filtering methods to recommend the food to the users. The application takes user ratings for different food items and stores it into the database. The application then recommends food items to the users on the basis of their ratings. In this project, we developed a website for a user -friendly interface which recommends the restaurant based on the choice of your interest. This is used for the users to predict the suitable and best restaurant as per their tastes. The content based filtering makes the recommendation more efficient so that each user can use this application for their easy prediction of the restaurant.

1.2 MOTIVATION

In the digital age, the restaurant industry has undergone a paradigm shift, with technology shaping how consumers discover and engage with dining options. The proliferation of online platforms has transformed how people select restaurants, placing immense importance on efficient and accurate recommendation systems. The motivation behind developing a restaurant recommendation system lies in addressing the challenges posed by the vast array of dining choices and the need to enhance user experience in this evolving landscape.

Consumers are inundated with an overwhelming number of dining options, often leading to decision fatigue and suboptimal choices. Traditional methods of restaurant discovery, such as word-of-mouth and random exploration, may not align with modern users' busy lifestyles and time constraints. Hence, a personalized and intelligent recommendation system becomes a necessity.

Furthermore, the proliferation of online reviews, ratings, and social media platforms has democratized the sharing of dining experiences. These user-generated content sources contain valuable insights that can be harnessed to provide tailored recommendations based on individual preferences, enhancing user satisfaction and loyalty.

The restaurant recommendation system caters to both users seeking new dining experiences and restaurant owners aiming to expand their reach. By harnessing machine learning techniques, the system serves as a bridge between users' preferences and a plethora of dining options, saving time and energy, while also supporting local businesses by driving footfall to their establishments.

In a nutshell, the motivation for this project stems from the need to streamline the restaurant selection process, provide users with personalized dining suggestions, and support local eateries through intelligent technology. The fusion of data science and dining experiences is poised to revolutionize how users discover, engage with, and relish their culinary journeys.

1.3 STATEMENT OF THE PROBLEM

In a world teeming with gastronomic options, selecting the perfect restaurant can sometimes feel like navigating a labyrinth without a map. The dining experience, once a straightforward affair, is now a complex decision-making process influenced by factors such as cuisine, ratings, costs, and personal preferences. As technology evolves, so do our expectations. Enter the challenge: how can we simplify this culinary conundrum and enhance the way people discover dining gems tailored to their tastes?
The age-old reliance on recommendations from friends and family, though valuable, has limitations in a digital age. Thus, we embark on a mission to revolutionize restaurant selection by engineering a sophisticated solution that harnesses the power of data and intelligence. Our goal is to create a restaurant recommendation system that serves as a personal dining concierge, offering tailored suggestions that align with individual palates, budgets, and moods.

The problems are clear:

1. Complex Dining Landscape: In today's world, the plethora of dining options poses a challenge for individuals seeking the perfect restaurant that aligns with their preferences and budgets.

2. Information Overload: The availability of extensive reviews, varying ratings, and diverse cuisines often overwhelms diners, leading to decision fatigue

3. Inefficient Recommendations: Traditional methods of seeking recommendations from friends or browsing through online reviews have limitations in providing personalized and accurate suggestions.

4. Restaurant Competitiveness: For restaurant owners, standing out in a competitive market and effectively reaching their target audience is an ongoing challenge.

5. Need for Efficient Solutions: As technology advances, there's a growing need for innovative solutions that leverage data and intelligence to streamline the dining decision-making process.

6. Personalized Recommendations: The solution intends to harness the power of machine learning and data analytics to provide personalized recommendations based on factors such as cuisine preferences, average ratings, and cost considerations.

7. User-Centered Design: The project's focus lies in building a user-centric platform that intuitively guides users through the dining discovery journey, considering their individual tastes and budgets.

8. Revolutionizing Dining Choices: Through the fusion of technology and culinary experiences, the project aims to redefine how people select restaurants, transforming it into an intuitive and exciting adventure.

1.4 OBJECTIVE

The main objective of this project is to design and develop a restaurant recommendation system that employs machine learning techniques to assist users in making informed and satisfying dining choices. The system aims to provide personalized restaurant suggestions based on various factors, including user preferences, average ratings, and cost considerations. The specific objectives of the project are as follows:

1. Data Collection and Preprocessing: Gather and preprocess restaurant data from sources such as the Zomato dataset. Cleanse and structure the data to ensure accuracy and consistency for subsequent analysis.

2. Feature Engineering: Create relevant features, such as the "Mean Rating," to quantify restaurant quality. Normalize these features using techniques like MinMaxScaler to ensure consistency across the dataset.

3. Cosine Similarity Calculation: Calculate the cosine similarity between restaurants using their TF-IDF vectors derived from customer reviews. Construct a similarity matrix to quantify the resemblance between establishments.

4. Recommendation Functions: Develop recommendation functions based on multiple criteria, including cuisine type, mean rating, and cost. Customize these functions to deliver restaurant suggestions that match user preferences.

5. Deployment: Deploy the recommendation system on a web platform using Flask and Streamlit. Allow users to interact with the system, input their preferences, and receive tailored restaurant recommendations.

6. Evaluation and Validation: Assess the effectiveness of the recommendation system by measuring the relevance and accuracy of the suggested restaurants against user preferences and reviews.

7. User Experience Enhancement: Enhance the user experience by providing clear and comprehensible recommendations, presenting results in an organized manner, and minimizing response times.

8. Local Business Support: Facilitate the discovery of local restaurants by highlighting their offerings in line with user preferences. Contribute to the growth of local businesses by connecting them with potential customers.

9. Documentation: Prepare comprehensive documentation that outlines the project's methodology, data preprocessing, feature engineering, recommendation algorithms, and deployment procedures.

10. Future Enhancements: Identify opportunities for further system refinement, including incorporating user feedback, exploring advanced recommendation algorithms, and expanding to accommodate additional user preferences.

1.5 FEASIBILITY STUDY

A feasibility study assesses the viability of a project in terms of technical, economic, operational, and schedule aspects. In the context of the restaurant recommendation system project, a feasibility study was conducted to evaluate the project's practicality and potential success. Here's a breakdown of the feasibility study:

1. Technical Feasibility:

- The project leverages established machine learning and natural language processing techniques, which are well-documented and supported by libraries like scikit-learn and NLTK.

- Adequate hardware and software resources, such as a standard computer and Python development environment, are readily available for the project's implementation.

- The chosen Flask and Streamlit frameworks are suitable for building web applications and deploying the recommendation system online.

2. Economic Feasibility:

- The project's core components can be developed using open-source tools and libraries, eliminating the need for costly software licenses.

- While there may be initial development costs, the system's long-term operational costs are minimal, mainly involving web hosting and maintenance.

3. Operational Feasibility:

- The project aligns with the growing trend of using technology to enhance daily experiences, making it relevant and appealing to users.

- The recommendation system's ease of use, interactive interface, and personalized suggestions contribute to its operational feasibility.

4. Schedule Feasibility:

- A well-structured project plan with clear milestones and deadlines was established at the project's outset.

- The project's modular design allows for parallel development of various components, accelerating the overall progress.

5. Legal and Ethical Considerations:

- The project uses publicly available restaurant data from the Zomato dataset, adhering to ethical guidelines and respecting user privacy.

6. Conclusion:

- The feasibility study indicates that the project is technically achievable, economically viable, and operationally and ethically sound.

- The project's scope, resources, and timeline align with the project team's capabilities and objectives.

1.6 SIGNIFICANCE OF PROJECT

The significance of this project lies in its capacity to revolutionize the way individuals discover and choose dining establishments, resulting in a more streamlined and enjoyable culinary experience. With the proliferation of dining options in today's urban landscapes, individuals are often confronted with the overwhelming task of selecting a restaurant that aligns with their preferences. This project addresses this issue by introducing an advanced recommendation system that combines various factors such as reviews, ratings, cuisines, and costs to offer tailored suggestions to users.

The project's primary goal is to alleviate decision fatigue and information overload that users commonly face when choosing a restaurant. By leveraging machine learning techniques and natural language processing, the recommendation system discerns patterns in user preferences and behavior, allowing it to generate highly relevant and personalized restaurant recommendations. This not only simplifies the decision-making process but also enhances the dining experience by introducing users to restaurants they might not have considered on their own.

Central to the project's significance is the creation of the "Mean Rating" feature. This feature aggregates user ratings for each restaurant, providing a comprehensive and standardized measure of its quality. The normalization of ratings using the MinMaxScaler ensures that the "Mean Rating" accurately reflects a restaurant's relative quality accurately. Consequently, users can make more informed decisions about where to dine, fostering a sense of trust and confidence in the recommendations provided by the system.

In the context of today's fast-paced lifestyles, the project's significance becomes even more apparent. As individuals juggle various commitments, finding the time to research and select a restaurant can be challenging. The recommendation system's ability to rapidly generate relevant suggestions caters to users' time constraints while also enhancing the likelihood of a satisfying dining experience. Furthermore, the interactive interface of the system contributes to its user-friendliness, making the process of exploring restaurant options engaging and enjoyable.

From a broader perspective, the project aligns with the evolving landscape of technology-driven solutions that enhance everyday experiences. As individuals increasingly rely on digital platforms for various aspects of their lives, the integration of advanced recommendation systems into the dining domain reflects a larger trend of harnessing technology to optimize decision-making and elevate user satisfaction.

In conclusion, the significance of this project lies in its ability to transform the way individuals engage with the dining landscape. By offering personalized and data-driven restaurant recommendations, the system empowers users to make informed choices while fostering an exploration of diverse culinary offerings. As a result, the project contributes to an enhanced dining experience, streamlined decision-making, and an increased sense of satisfaction for users seeking to discover new and exciting restaurant options.

1.7 BENEFICIARY OF THE SYSTEM

The recommendation system developed in this project benefits a diverse range of individuals who engage in dining-out experiences. The system caters to the needs and preferences of food enthusiasts, busy professionals, families, tourists, and anyone seeking culinary exploration. Each of these beneficiary groups can reap specific advantages from the recommendation system:

1. Food Enthusiasts: Culinary aficionados who have a keen interest in exploring different cuisines and discovering hidden culinary gems can benefit from the system's ability to suggest unique and diverse restaurant options. The personalized recommendations align with their adventurous palate, allowing them to indulge in novel dining experiences.

2. Busy Professionals: Professionals with limited time to research and select dining options can rely on the system's efficient recommendations. By eliminating the need for extensive manual research, they can quickly make informed decisions and enjoy quality meals without the stress of choice overload.

3. Families: Families seeking suitable dining options that cater to varying tastes and preferences can utilize the system to find restaurants that offer a range of cuisines and price points. This streamlines the process of selecting a family-friendly restaurant that caters to everyone's requirements.

4. Tourists: Travelers exploring new cities or regions can benefit from the system's local restaurant recommendations. By factoring in ratings, reviews, and cuisines, tourists can discover authentic dining experiences that align with their preferences and the local culture.

5. Adventurous Eaters: Individuals open to trying new cuisines and flavors can rely on the system to suggest restaurants that match their adventurous spirit. By considering factors like cuisines and reviews, the system encourages them to broaden their culinary horizons.

6. Cost-Conscious Individuals: Those mindful of their budget can benefit from the system's cost-based recommendations. By factoring in cost ranges, the system ensures that users receive suggestions that align with their financial considerations.

7. Restaurant Owners: Restaurant owners and managers can indirectly benefit from the system's recommendations. Positive reviews and ratings from users could lead to increased footfall and popularity, thus enhancing the restaurant's reputation and business.

In essence, the recommendation system's beneficiaries encompass a diverse spectrum of individuals seeking convenience, personalization, and enhanced dining experiences. By catering to various user profiles and preferences, the system enhances the overall dining journey and contributes to a more satisfying and enjoyable exploration of the culinary landscape.

2.1 LITERATURE REVIEW

The development of restaurant recommendation systems has garnered substantial attention due to its potential to enhance user dining experiences and drive business growth in the food and hospitality industry. Various research studies have explored different aspects of recommendation systems, catering to diverse user needs. While the current project focuses on machine learning-based recommendations using attributes like ratings, reviews, cuisines, and costs, it's valuable to highlight related work in this domain:

1. Collaborative Filtering: Collaborative filtering techniques, a widely researched approach, involve recommending items based on the preferences of similar users. Research by Sarwar et al. (2001) demonstrated the effectiveness of collaborative filtering in recommending restaurants to users based on historical preferences.

2. Content-Based Filtering: Content-based filtering leverages item attributes to recommend similar items. Baltrunas et al. (2009) explored the use of content-based recommendations for restaurants, considering attributes like cuisines, locations, and user preferences.

3. Hybrid Approaches: Hybrid recommendation systems combine multiple techniques to provide more accurate and diverse suggestions. A study by Adomavicius and Tuzhilin (2005) investigated hybrid methods, which could combine collaborative filtering and content-based methods for better results.

4. Natural Language Processing: While not directly used in this project, some recommendation systems use natural language processing (NLP) to analyze reviews and feedback. Researchers like Zhao et al. (2018) have delved into using NLP techniques to extract sentiment and feature information from textual reviews for better recommendations.

5. Personalization and Diversity: Ensuring diverse recommendations is crucial for user satisfaction. Research by Zhang et al. (2019) focused on incorporating diversity-aware mechanisms into recommendation systems to prevent over-representing popular items and introducing novel options to users.

6. Deployment and User Interface: Deploying recommendation systems on web platforms is an active area of research. Projects like this often use frameworks like Flask and Streamlit to provide user-friendly interfaces. The work by Carvalho et al. (2020) discussed building a web-based recommendation system for restaurants using Flask.

7. Ethical Considerations: As recommendation systems influence user choices, ethical concerns arise. Studies like Abdollahpouri et al. (2019) highlight the importance of transparency, fairness, and user control in designing recommendation systems.

In summary, the literature underscores the diversity of recommendation techniques, ranging from collaborative and content-based filtering to hybrid approaches and NLP-driven sentiment analysis. The existing research provides a solid foundation for building effective recommendation systems that cater to various user preferences and enhance their dining experiences.

2.2 RELATED WORK

In the context of restaurant recommendation systems, various approaches have been explored to provide users with relevant dining suggestions. The following related works highlight methodologies that align with the core features of our recommendation system, including review, rating, cuisine, and cost-based recommendations.

1. Collaborative Filtering: Collaborative filtering techniques leverage user interactions and preferences to provide recommendations. User-based and item-based collaborative filtering analyze historical user-item interactions for personalized suggestions.

2. Content-Based Filtering: Content-based filtering suggests items based on their attributes, such as cuisine, price, and ratings. This approach is effective in catering to specific user preferences.

3. Hybrid Approaches: Hybrid methods combine collaborative and content-based techniques for improved recommendation accuracy. They leverage the strengths of both approaches to provide more diverse and relevant suggestions.

4. Matrix Factorization: Matrix factorization methods capture latent factors in user-item interactions to generate recommendations.

5. Mean Rating Normalization: Similar to the project, normalizing mean ratings using techniques like Min-Max scaling ensures fair comparison and accurate recommendations.

6. Cost-Based Recommendations: Recommending restaurants based on cost preferences is valuable for users looking for specific budget-friendly options.

7. Review and Rating Analysis: Incorporating user reviews and ratings can enhance recommendations by considering user sentiments and feedback.

8. Location-Agnostic Recommendations: Recommendations that do not rely on location information are relevant for scenarios where users are seeking options beyond their current location.

9. Real-Time Adaptation: Systems that adapt in real-time to changing user preferences and trends enhance user satisfaction and engagement.

3.1 METHODOLOGY

The purpose of this project is to develop a restaurant recommendation system that assists users in discovering similar restaurants based on various criteria such as cuisine, mean rating, and cost. The goal is to provide personalized and relevant dining suggestions, enhancing the user's dining experience and aiding their decision-making process.

The restaurant recommendation system employs various criteria such as cuisine, mean rating, and cost to suggest restaurants with similar attributes to those specified by the user. It uses techniques like TF-IDF for text analysis and cosine similarity for comparison. By inputting a restaurant name or specific criteria, users can receive a list of top similar restaurants, aiding them in discovering dining options that match their preferences and enhancing their overall restaurant selection experience.

Step 1: Data Collection and Preprocessing

1. Data Source and Attributes: The project utilizes the Zomato dataset of the cities Bangalore and Delhi from Kaggle.

For Bangalore:
URL - contains the URL of the restaurant on the zomato website
address - contains the address of the restaurant in Bengaluru
name - contains the name of the restaurant
online_order - whether online ordering is available in the restaurant or
not *book_table* - table book option available or not
rate - contains the overall rating of the restaurant out of 5
votes - contains total number of rating for the restaurant as of the above-mentioned date
phone - contains the phone number of the restaurant
location - contains the neighborhood in which the restaurant is located
rest_type - restaurant type
dish_liked - dishes people liked in the restaurant
cuisines - food styles, separated by a comma
approx_cost(for two people) - contains the approximate cost for a meal for two people
reviews_list - list of tuples containing reviews for the restaurant, each tuple consists of two values

For Delhi:
Restaurant_Name - Holds the name of the Restaurant
Category - Type of Food the restaurant/cafe is serving.
Pricing_for_2 - The price of ordering/dine-in for 2 people
Locality - The locality/street where the restaurant is situated.
Dining_Rating - The average rating for the restaurant given by people who dine-in
Dining_Review_Count - Count of Dining_Ratings given.
Delivery_Rating - The average rating is given by people who order food online from the restaurant
Delivery_Rating_Count - Count of Delivery Ratings given.
Website - The Zomato's URL for that particular restaurant
Address - Street Address for the restaurant.
Phn_no - Restaurant's Phone Number as listed on Zomato.

Latitude - Geographic Latitude coordinates for the restaurant.
Longitude - Geographic Longitude coordinates for the restaurant.
Known_For2 - What Restaurant is Famous for? (Ambience / Food) (Column 1)
Known_For22 - What Restaurant is Famous for? (Ambience / Food) (Column 2)

2. Data Loading and Preprocessing: The dataset is loaded using Pandas from a CSV file. Preprocessing involves removing irrelevant columns like 'url', 'rest_type' and 'dish_liked'. Ratings are cleaned, and costs are converted to numeric values.

3. Data Cleaning Techniques: Missing values in columns like 'approx_cost' are handled by dropping or imputing them based on context. Columns are renamed for clarity and consistency. Duplicate rows are removed to avoid redundancy.

4. Data Transformation: Ratings are normalized using Min-Max scaling to bring them within a consistent range. Costs are converted to numeric by removing commas. Ratings' text part is removed to convert them to numerical values for analysis. This facilitates better comparison and recommendation calculation.

Step 2: Text Preprocessing

1. Lowercasing: Converting all text to lowercase ensures uniformity in the dataset. This avoids case-related discrepancies during analysis and recommendation.

2. Punctuation Removal: Eliminating punctuation symbols from text reduces noise and simplifies the data. Punctuation doesn't carry significant meaning and can hinder analysis.

3. Stopword Removal: Removing common words like "and," "the," etc., which don't contribute much to the context, helps reduce the dimensionality of the data. This speeds up processing and enhances relevance.

4. URL Removal: Extracting URLs from text ensures they don't interfere with analysis. URLs usually don't add any value to restaurant recommendations.

Step 3: Feature Engineering

1. Creation of "Mean Rating" Feature: The "Mean Rating" feature is created by calculating the average rating of each restaurant based on its individual ratings. It involves iterating through all restaurants, calculating their average rating, and storing it as a new feature in the dataset. This feature represents the overall quality or popularity of a restaurant based on its ratings.

2. Normalization using MinMaxScaler: After calculating the mean ratings, the values are normalized using MinMaxScaler. This technique scales the values to a specific range (usually [0, 1]) to maintain uniformity and ensure that all ratings are comparable. In this case, the range is adjusted to [1, 5] to retain the original rating scale.

Step 4: Cosine Similarity Calculation

1. Cosine Similarity Overview: Cosine similarity, a measure of similarity between two vectors, determines how closely two restaurants align in terms of features. It's computed as the cosine of the angle between the vectors, indicating their similarity.

2. TF-IDF Vector Creation: The reviews are converted into TF-IDF (Term Frequency-Inverse Document Frequency) vectors. This process captures the importance of words in reviews, taking into account both their frequency in a restaurant's reviews and their rarity across all restaurants.

3. Cosine Similarity Matrix: Using the TF-IDF vectors, a cosine similarity matrix is constructed. This matrix calculates the similarity between each pair of restaurants. Higher values indicate greater similarity, aiding in recommending similar restaurants based on review content.

Step 5: <u>Recommendation Functions</u>

1. Recommendation Function Creation: Multiple recommendation functions were developed based on cuisine, mean rating, and cost. Each function takes input parameters and returns similar restaurant suggestions.
2. Function Parameters: Parameters include the cuisine type for the cuisine-based recommendation, the restaurant's name for mean rating and cost-based recommendations, and an optional parameter for the number of top recommendations. These parameters influence the results and specificity of recommendations.

3. Logic for Selection and Presentation: For cuisine recommendations, the normalized cuisine value is used to retrieve similar restaurants. For mean rating and cost recommendations, restaurants with similar mean rating or cost values are selected. Results are sorted and presented, aiding users in discovering restaurants with similar attributes.

Step 6: <u>Frontend</u>

The frontend is responsible for providing a user-friendly interface for users to input their preferences and receive restaurant recommendations.

1. User Interface (UI): Develop the user interface using React.js. Design input forms where users can specify their preferences such as cuisine, price range, etc.

2. User Interaction: Capture user inputs from the UI components and send them to the backend for processing. This can be achieved through HTTP requests (POST/GET).

3. Data Presentation: Display the received restaurant recommendations in a visually appealing manner. This could involve listing the top recommended restaurants, displaying images, ratings, and relevant details.

Step 7: <u>Backend</u>

The backend handles user requests, interacts with the machine learning model, and sends back the recommendations to the frontend.

1. API Endpoints: Create Flask routes (API endpoints) to handle incoming requests from the frontend. For instance, you might have an endpoint to receive user preferences and another to send back restaurant recommendations.

2. Request Processing: When receiving user preferences, validate and sanitize the input data. Then, pass the data to the machine learning model for recommendation generation.

3. Machine Learning Interaction: Load the trained machine learning model from the pickle file. Utilize this model to generate restaurant recommendations based on the provided user preferences.

4. Response Formatting: Once recommendations are generated by the model, format them into a suitable structure, that is JSON to be sent back as a response to the frontend.

Conclusion

The project successfully developed a restaurant recommendation system using text data and numerical features. It effectively suggests similar restaurants based on cuisine, mean rating, and cost.

The recommendation system offers relevant suggestions to users, aiding them in discovering restaurants with similar attributes. Users can explore options that match their preferences.

The system could benefit from incorporating more advanced techniques like collaborative filtering and hybrid approaches. Integrating user feedback and enhancing the cosine similarity metric can improve accuracy. Additionally, expanding the dataset and optimizing the preprocessing steps could yield better results.

4.1 IMPLEMENTATION (Restaurant Recommendation System-Bangalore)

In [24]:
```
#Before Text Processing
data[['reviews_list', 'cuisines']].head()
```
Out[24]:

	reviews_list	cuisines
0	[('Rated 4.0', 'RATED\n A beautiful place to ...	North Indian, Mughlai, Chinese
1	[('Rated 4.0', 'RATED\n Had been here for din...	Chinese, North Indian, Thai
2	[('Rated 3.0', "RATED\n Ambience is not that ...	Cafe, Mexican, Italian
3	[('Rated 4.0', "RATED\n Great food and proper...	South Indian, North Indian
4	[('Rated 4.0', 'RATED\n Very good restaurant ...	North Indian, Rajasthani

In [25]:
```
#Lower Casing
data["reviews_list"] = data["reviews_list"].str.lower()
data[['reviews_list', 'cuisines']].head()
```
Out[25]:

	reviews_list	cuisines
0	[('rated 4.0', 'rated\n a beautiful place to ...	North Indian, Mughlai, Chinese
1	[('rated 4.0', 'rated\n had been here for din...	Chinese, North Indian, Thai
2	[('rated 3.0', "rated\n ambience is not that ...	Cafe, Mexican, Italian
3	[('rated 4.0', "rated\n great food and proper...	South Indian, North Indian
4	[('rated 4.0', 'rated\n very good restaurant ...	North Indian, Rajasthani

In [26]:
```
#Removal of Punctuations
import string
punc = string.punctuation
def remove_punctuation(text):
    return text.translate(str.maketrans('', '', punc))

data["reviews_list"] = data["reviews_list"].apply(lambda text: remove_punctuation(text)
data[['reviews_list', 'cuisines']].head()
```
Out[26]:

	reviews_list	cuisines
0	rated 40 ratedn a beautiful place to dine int...	North Indian, Mughlai, Chinese
1	rated 40 ratedn had been here for dinner with...	Chinese, North Indian, Thai
2	rated 30 ratedn ambience is not that good eno...	Cafe, Mexican, Italian
3	rated 40 ratedn great food and proper karnata...	South Indian, North Indian
4	rated 40 ratedn very good restaurant in neigh...	North Indian, Rajasthani

In [27]:
```
#Removal of Stopwords
Stopwords = set(stopwords.words('english'))
def remove_stopwords(text):
    return " ".join([word for word in str(text).split() if word not in Stopwords])

data["reviews_list"] = data["reviews_list"].apply(lambda text: remove_stopwords(text))
data[['reviews_list', 'cuisines']].head()
```
Out[27]:

	reviews_list	cuisines
0	rated 40 ratedn beautiful place dine inthe int...	North Indian, Mughlai, Chinese
1	rated 40 ratedn dinner family turned good choo...	Chinese, North Indian, Thai
2	rated 30 ratedn ambience good enough pocket fr...	Cafe, Mexican, Italian
3	rated 40 ratedn great food proper karnataka st...	South Indian, North Indian
4	rated 40 ratedn good restaurant neighbourhood ...	North Indian, Rajasthani

In [28]:
```
#Removal of URLS
def remove_urls(text):
    url_pattern = re.compile(r'https?://\S+|www\.\S+')
    return url_pattern.sub(r'', text)

data["reviews_list"] = data["reviews_list"].apply(lambda text: remove_urls(text))
data[['reviews_list', 'cuisines']].head()
```
Out[28]:

	reviews_list	cuisines
0	rated 40 ratedn beautiful place dine inthe int...	North Indian, Mughlai, Chinese
1	rated 40 ratedn dinner family turned good choo...	Chinese, North Indian, Thai
2	rated 30 ratedn ambience good enough pocket fr...	Cafe, Mexican, Italian
3	rated 40 ratedn great food proper karnataka st...	South Indian, North Indian
4	rated 40 ratedn good restaurant neighbourhood ...	North Indian, Rajasthani

In [31]:
```
data=data.drop(['address','rest_type', 'type', 'menu_item', 'votes'],axis=1)
```

Recommendation by Reviews

In [50]:
```
df_percent = data.sample(frac=0.5)
df_percent.set_index('name', inplace=True)
indices = pd.Series(df_percent.index)
```

In [51]:
```
tfidf = TfidfVectorizer(analyzer='word', ngram_range=(1, 2), min_df=1, stop_words='engl
tfidf_matrix = tfidf.fit_transform(df_percent['reviews_list'])
cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
```

In [73]:
```
def recommend_like(name, top_n=-1, cosine_similarities=cosine_similarities):
    recommend_restaurant = []
    try:
        idx = indices[indices == name].index[0]
        score_series = pd.Series(cosine_similarities[idx])
        top_indexes = list(score_series.iloc[1:-1].index)

        for each in top_indexes:
            recommend_restaurant.append(list(df_percent.index)[each])

        df_new = pd.DataFrame(columns=['cuisines', 'Mean Rating', 'cost'])

        for each in recommend_restaurant:
            selected_data = df_percent[['cuisines', 'Mean Rating', 'cost']][df_percent.
            df_new = pd.concat([df_new, selected_data])

        df_new = df_new.drop_duplicates(subset=['cuisines', 'Mean Rating', 'cost'], kee
        df_new = df_new.sort_values(by='Mean Rating', ascending=False).head(top_n)

        print(f'TOP RESTAURANTS LIKE {name.upper()}: ')
        df_new.reset_index(drop=True, inplace=True)
        return df_new
    except:
        print('No restaurants with similar reviews.')
```

In [63]:
```
recommend_like('Jalsa')
```

```
TOP RESTAURANTS LIKE JALSA:
```

Out[63]:

	cuisines	Mean Rating	cost
0	Continental, North Indian, Italian, South Indi...	5.00	1600.0
1	BBQ	4.81	1300.0
2	South Indian	4.74	100.0
3	Chinese, Continental, North Indian, Finger Food	4.68	1200.0
4	Desserts, Fast Food	4.64	350.0
...	...	...	...
1535	Pizza, Fast Food	1.95	700.0
1536	South Indian	1.90	300.0
1537	North Indian	1.90	500.0
1538	North Indian, Chinese, Fast Food, Rolls, Juices	1.90	400.0
1539	Bakery, Fast Food	1.77	500.0

1540 rows × 3 columns

Recommendation by Cuisine

In [64]:
```
df_percent = data.sample(frac=1)
df_percent.set_index('cuisines'.lower(), inplace=True)
indices = pd.Series(df_percent.index)

from fuzzywuzzy import fuzz

cuisine_mapping = {}
for index, row in data.iterrows():
    for cuisine in row['cuisines'].split(', '):
        normalized_cuisine = cuisine.lower()
        if normalized_cuisine not in cuisine_mapping:
            cuisine_mapping[normalized_cuisine] = []
        cuisine_mapping[normalized_cuisine].append(index)

def recommend_by_cuisine(criteria_value, top_n=-1, cosine_similarities=cosine_similarit
    recommend_restaurant = []
    normalized_criteria = criteria_value.lower()
    matching_restaurants = cuisine_mapping.get(normalized_criteria, [])

    for restaurant_index in matching_restaurants:
        recommend_restaurant.append(restaurant_index)

    df_new = pd.DataFrame(columns=['name', 'cuisines', 'Mean Rating', 'cost'])
    for each in recommend_restaurant:
        selected_data = data[['name', 'cuisines', 'Mean Rating', 'cost']][data.index ==
        df_new = pd.concat([df_new, selected_data])
    df_new = df_new.drop_duplicates(subset=['name', 'cuisines', 'Mean Rating', 'cost'],
    df_new = df_new.sort_values(by='Mean Rating', ascending=False).head(top_n)

    print(f'TOP RESTAURANTS WITH SIMILAR CUISINE: ')
    df_new.reset_index(drop=True, inplace=True)
    return df_new
```

In [65]:
```
recommend_by_cuisine('North Indian')
```

```
TOP RESTAURANTS WITH SIMILAR CUISINE:
```

Out[65]:

	name	cuisines	Mean Rating	cost
0	The Reservoire	Continental, North Indian, Chinese, American, ...	4.56	1300.0
1	Levitate Brewery And Kitchen	Finger Food, North Indian, Continental	4.48	1500.0
2	Output Bengaluru	North Indian, Continental	4.48	1000.0
3	Feast - Sheraton Grand Bengaluru Whitefield Ho...	Continental, Asian, South Indian, North Indian	4.48	2500.0
4	Saffron - Radisson Blu	North Indian, Mughlai, Lucknowi	4.48	2500.0
...	...	...	...	...
435	Sukhi'S Restaurant	North Indian, Chinese, Momos	2.16	250.0
436	Nightowl	North Indian	2.16	400.0
437	Reddy'S Hyderabadi Biryani	Chinese, North Indian, South Indian, Biryani, ...	2.16	500.0
438	Navya'S	North Indian, Andhra	2.08	600.0
439	Gongura	Andhra, North Indian, South Indian	2.03	500.0

440 rows × 4 columns

Recommendation by Rating

In [66]:
```
df_percent = data.sample(frac=1)
df_percent.set_index('Mean Rating', inplace=True)
indices = pd.Series(df_percent.index)

def recommend_by_mean_rating(mean_rating_value, top_n=-1):
    recommend_restaurant = data[data['Mean Rating'] >= mean_rating_value].index
    df_new = data.loc[recommend_restaurant, ['name', 'cuisines', 'Mean Rating', 'cost']]
    df_new = df_new.sample(frac=1)
    df_new = df_new.head(top_n)
    print(f'TOP RESTAURANTS WITH A MINIMUM RATING OF {mean_rating_value}: ')
    df_new.reset_index(drop=True, inplace=True)
    return df_new
```

In [46]:
```
recommend_by_mean_rating(4)
```

TOP 10 RESTAURANTS WITH A MINIMUM RATING OF 4:

Out[46]:

name	cuisines	Mean Rating	cost
Birinz	Biryani, Kebab, Fast Food	4.07	650.0
Mojo Pizza - 2X Toppings	Pizza	4.13	600.0
Gramin	North Indian	4.23	600.0
Skoolroom	Cafe, Continental, Italian, Burger, Beverages	4.23	700.0
Murphy'S Brewhouse - The Paul Bangalore	Continental, North Indian, Finger Food	4.23	1700.0
Smally'S Resto Cafe	Cafe, Italian, Burger, American, Steak	4.06	650.0
Mama Mia!	Desserts, Cafe	4.04	400.0
Cafe Medley	Cafe, Continental, Burger	4.10	800.0
Smally'S Resto Cafe	Cafe, Italian, Burger, American, Steak	4.06	650.0
Lot Like Crepes	Cafe, Desserts, Continental	4.63	550.0

Recommendation by Cost

In [68]:
```
df_percent = data.sample(frac=1)
df_percent.set_index('cost', inplace=True)
indices = pd.Series(df_percent.index)

def recommend_by_cost(min_value,max_value, top_n=-1):
    recommend_restaurant = data[(data['cost'] > min_value) & (data['cost'] <= max_value
    df_new = data.loc[recommend_restaurant, ['name', 'cuisines', 'Mean Rating', 'cost']
    df_new = df_new.sample(frac=1)
    df_new = df_new.head(top_n)
    print(f'TOP RESTAURANTS IN THE PROVIDED RANGE: ')
    df_new.reset_index(drop=True, inplace=True)
    return df_new
```

In [69]:
```
recommend_by_cost(1000,2000)
```

TOP RESTAURANTS IN THE PROVIDED RANGE:

Out[69]:

	name	cuisines	Mean Rating	cost
0	One For The Road	Continental, North Indian, Finger Food	4.14	1300.0
1	Parika Multicuisine Restaurant	North Indian, Chinese, Mughlai	3.58	1200.0
2	Mainland China	Chinese	4.48	1700.0
3	Kailash Parbat	South Indian, Chinese, North Indian	2.73	1200.0
4	Nostradamus Bar & Lounge	North Indian, South Indian, Chinese, Continental	3.32	1500.0
...	...	...	...	...
4506	Om Made Cafe	Cafe, Continental, European, Juices	3.94	1300.0
4507	Le Rock	Continental, Chinese	3.89	2000.0
4508	The London Curry House - The Royale Senate Hotel	Pizza, North Indian, Modern Indian	4.48	1300.0
4509	Jalsa	North Indian, Mughlai	3.99	1500.0
4510	Churchill'S	North Indian, Continental	3.97	1100.0

4511 rows × 4 columns

IMPLEMENTATION (Restaurant Recommendation System-Delhi)

```
In [1]: import numpy as np
        import pandas as pd
        import warnings
        warnings.filterwarnings('ignore')
        import re
        from nltk.corpus import stopwords
        from sklearn.metrics.pairwise import linear_kernel
        from sklearn.feature_extraction.text import CountVectorizer
        from sklearn.feature_extraction.text import TfidfVectorizer
```

```
In [2]: data = pd.read_csv("/Users/triptibhardwaj/Downloads/DelhiNCR Restaurants.csv.xls")
        data.head()
```

Out[2]:

	Restaurant_Name	Category	Pricing_for_2	Locality	Dining_Rating	Dining_Review_Count	D
0	Rustom's	Parsi, Street Food, North Indian, Desserts, Be...	2100	ITO, New Delhi	4.9	1885	
1	Cafe Lota	Cafe, South Indian, North Indian, Beverages	1200	Pragati Maidan, New Delhi	4.9	3748	
2	Dum-Pukht - ITC Maurya	Mughlai, North Indian, Desserts	5000	ITC Maurya, Chanakyapuri, New Delhi	4.9	1371	
3	Burma Burma	Asian, Burmese, Bubble Tea, Desserts, Salad	1600	Cyber Hub, DLF Cyber City, Gurgaon	4.9	2636	
4	The Big Chill	Continental, American, Italian	1500	Khan Market, New Delhi	4.9	6487	

```
In [3]: print(data.isnull().sum())

        Restaurant_Name           0
        Category                  0
        Pricing_for_2             0
        Locality                  0
        Dining_Rating             0
        Dining_Review_Count       0
        Delivery_Rating         402
        Delivery_Rating_Count     0
        Website                   0
        Address                   0
        Phone_No                  0
        Latitude                  0
        Longitude                 0
        Known_For2              405
        Known_For22             841
        dtype: int64
```

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1965 entries, 0 to 1964
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Restaurant_Name        1965 non-null   object
 1   Category               1965 non-null   object
 2   Pricing_for_2          1965 non-null   int64
 3   Locality               1965 non-null   object
 4   Dining_Rating          1965 non-null   float64
 5   Dining_Review_Count    1965 non-null   int64
 6   Delivery_Rating        1563 non-null   float64
 7   Delivery_Rating_Count  1965 non-null   int64
 8   Website                1965 non-null   object
 9   Address                1965 non-null   object
 10  Phone_No               1965 non-null   object
 11  Latitude               1965 non-null   float64
 12  Longitude              1965 non-null   float64
 13  Known_For2             1560 non-null   object
 14  Known_For22            1124 non-null   object
dtypes: float64(4), int64(3), object(8)
memory usage: 230.4+ KB
```

In [5]:
```
data=data.drop(['Dining_Review_Count','Delivery_Rating','Delivery_Rating_Count','Websit
```

In [6]:
```
data.duplicated().sum()
data.drop_duplicates(inplace=True)
data.head()
```

Out[6]:

	Restaurant_Name	Category	Pricing_for_2	Locality	Dining_Rating	Known_For22
0	Rustom's	Parsi, Street Food, North Indian, Desserts, Be...	2100	ITO, New Delhi	4.9	NaN
1	Cafe Lota	Cafe, South Indian, North Indian, Beverages	1200	Pragati Maidan, New Delhi	4.9	Artistic Decor, The Service, Natural Ambience,...
2	Dum-Pukht - ITC Maurya	Mughlai, North Indian, Desserts	5000	ITC Maurya, Chanakyapuri, New Delhi	4.9	NaN
3	Burma Burma	Asian, Burmese, Bubble Tea, Desserts, Salad	1600	Cyber Hub, DLF Cyber City, Gurgaon	4.9	Knowledgeable Staff, Authentic, Soothing Ambie...
4	The Big Chill	Continental, American, Italian	1500	Khan Market, New Delhi	4.9	Retro Ambience, Yummy Desserts, Big Portions, ...

In [7]:
```
data.isnull().sum()
```

Out[7]:
```
Restaurant_Name      0
Category             0
Pricing_for_2        0
Locality             0
Dining_Rating        0
Known_For22        841
dtype: int64
```

In [8]:
```
data.dropna(how='any',inplace=True)
data.head()
```

Out[8]:

	Restaurant_Name	Category	Pricing_for_2	Locality	Dining_Rating	Known_For22
1	Cafe Lota	Cafe, South Indian, North Indian, Beverages	1200	Pragati Maidan, New Delhi	4.9	Artistic Decor, The Service, Natural Ambience,...
3	Burma Burma	Asian, Burmese, Bubble Tea, Desserts, Salad	1600	Cyber Hub, DLF Cyber City, Gurgaon	4.9	Knowledgeable Staff, Authentic, Soothing Ambie...
4	The Big Chill	Continental, American, Italian	1500	Khan Market, New Delhi	4.9	Retro Ambience, Yummy Desserts, Big Portions, ...
5	Carnatic Cafe	South Indian	600	Lodhi Colony, New Delhi	4.9	Cosy Ambience, Live Kitchen, Quiet Place, Calm...
6	Cocktails & Dreams, Speakeasy	Nepalese, Tibetan, Beverages	2500	Sector 15, Gurgaon	4.9	Jazz Music, Band Playing, Bartender, Live Band...

In [9]:
```
data = data.rename(columns={'Restaurant_Name':'name','Category':'cuisines','Pricing_for
```

In [10]:
```
data.name = data.name.apply(lambda x:x.title())
```

In [11]:
```
data['cost'] = data['cost'].astype(float)
```

In [12]:
```
restaurants = list(data['name'].unique())
data['Mean Rating'] = 0

for i in range(len(restaurants)):
    data['Mean Rating'][data['name'] == restaurants[i]] = data['rating'][data['name'] =
```

In [13]:
```
data.loc[data.name =='Burma Burma']
```

Out[13]:

	name	cuisines	cost	location	rating	reviews_list	Mean Rating
3	Burma Burma	Asian, Burmese, Bubble Tea, Desserts, Salad	1600.0	Cyber Hub, DLF Cyber City, Gurgaon	4.9	Knowledgeable Staff, Authentic, Soothing Ambie...	4.9

In [14]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (1,5))
data[['Mean Rating']] = scaler.fit_transform(data[['Mean Rating']]).round(2)
```

```
In [15]: data.head()
Out[15]:
```

	name	cuisines	cost	location	rating	reviews_list	Mean Rating
1	Cafe Lota	Cafe, South Indian, North Indian, Beverages	1200.0	Pragati Maidan, New Delhi	4.9	Artistic Decor, The Service, Natural Ambience,...	5.0
3	Burma Burma	Asian, Burmese, Bubble Tea, Desserts, Salad	1600.0	Cyber Hub, DLF Cyber City, Gurgaon	4.9	Knowledgeable Staff, Authentic, Soothing Ambie...	5.0
4	The Big Chill	Continental, American, Italian	1500.0	Khan Market, New Delhi	4.9	Retro Ambience, Yummy Desserts, Big Portions, ...	5.0
5	Carnatic Cafe	South Indian	600.0	Lodhi Colony, New Delhi	4.9	Cosy Ambience, Live Kitchen, Quiet Place, Calm...	5.0
6	Cocktails & Dreams, Speakeasy	Nepalese, Tibetan, Beverages	2500.0	Sector 15, Gurgaon	4.9	Jazz Music, Band Playing, Bartender, Live Band...	5.0

In [16]:
```
#Before Text Processing
data[['reviews_list', 'cuisines']].head()
```

Out[16]:

	reviews_list	cuisines
1	Artistic Decor, The Service, Natural Ambience,...	Cafe, South Indian, North Indian, Beverages
3	Knowledgeable Staff, Authentic, Soothing Ambie...	Asian, Burmese, Bubble Tea, Desserts, Salad
4	Retro Ambience, Yummy Desserts, Big Portions, ...	Continental, American, Italian
5	Cosy Ambience, Live Kitchen, Quiet Place, Calm...	South Indian
6	Jazz Music, Band Playing, Bartender, Live Band...	Nepalese, Tibetan, Beverages

In [17]:
```
#Lower Casing
data["reviews_list"] = data["reviews_list"].str.lower()

#Removal of Punctuations
import string
punc = string.punctuation
def remove_punctuation(text):
    return text.translate(str.maketrans('', '', punc))

data["reviews_list"] = data["reviews_list"].apply(lambda text: remove_punctuation(text)

#Removal of Stopwords
Stopwords = set(stopwords.words('english'))
def remove_stopwords(text):
    return " ".join([word for word in str(text).split() if word not in Stopwords])

data["reviews_list"] = data["reviews_list"].apply(lambda text: remove_stopwords(text))

#Removal of URLS
def remove_urls(text):
    url_pattern = re.compile(r'https?://\S+|www\.\S+')
    return url_pattern.sub(r'', text)

data["reviews_list"] = data["reviews_list"].apply(lambda text: remove_urls(text))
data[['reviews_list', 'cuisines']].head()
```

Out[17]:

	reviews_list	cuisines
1	artistic decor service natural ambience fusion...	Cafe, South Indian, North Indian, Beverages
3	knowledgeable staff authentic soothing ambienc...	Asian, Burmese, Bubble Tea, Desserts, Salad
4	retro ambience yummy desserts big portions big...	Continental, American, Italian
5	cosy ambience live kitchen quiet place calm co...	South Indian
6	jazz music band playing bartender live band li...	Nepalese, Tibetan, Beverages

Recommendation by Reviews

In [18]:
```
df_percent = data.sample(frac=1)
df_percent.set_index('name', inplace=True)
indices = pd.Series(df_percent.index)
```

In [19]:
```
tfidf = TfidfVectorizer(analyzer='word', ngram_range=(1, 2), min_df=1, stop_words='engl
tfidf_matrix = tfidf.fit_transform(df_percent['reviews_list'])
cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
```

6.1 CONCLUSION

In conclusion, this project has successfully developed a restaurant recommendation system that utilizes machine learning techniques to provide personalized dining suggestions to users. By incorporating attributes such as ratings, reviews, cuisines, and costs, the system aims to enhance user experiences by offering relevant and diverse restaurant options. The significance of the project lies in its ability to aid users in discovering new dining establishments that align with their preferences and constraints.

Through rigorous data preprocessing, including text normalization and feature engineering, the system has effectively transformed raw data into meaningful insights. The utilization of TF-IDF vectors and cosine similarity calculations further ensures accurate recommendations by capturing the semantic context of user reviews.

The recommendation functions developed for cuisine, mean rating, and cost provide users with multiple avenues to explore and discover restaurants that match their criteria. The user-friendly interface, facilitated by frameworks like Flask and Streamlit, enables seamless interaction and enhances usability.

The project's feasibility study underscores the technical, economic, operational, and schedule viability of the recommendation system. It operates efficiently within available resources and aligns with the growing trend of using technology to improve everyday experiences.

While the system's effectiveness is evident from its recommendations, continuous improvement remains essential. Future enhancements could involve incorporating user feedback to fine-tune the recommendation algorithms, leveraging sentiment analysis for a deeper understanding of user preferences, and integrating real-time data for more up-to-date suggestions.

In conclusion, this restaurant recommendation system serves as a valuable tool for users seeking personalized dining options and contributes to the evolution of technology-driven enhancements in the food and hospitality industry. The successful implementation of this project encourages further exploration and innovation in recommendation system development.

REFERENCES

I. Abdollahpouri, H., Burke, R., & Mobasher, B. (2019). *Fairness-aware social recommendation*. In Proceedings of the 2019 ACM Conference on Recommender Systems (pp. 192-200).

II. Adomavicius, G., & Tuzhilin, A. (2005). *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749.

III. Baltrunas, L., Makcinskas, T., & Ricci, F. (2009). *Group recommendations with rank aggregation and collaborative filtering.* In Proceedings of the 2009 ACM Conference on Recommender Systems (pp. 119-126).

IV. Carvalho, R. S., Felizardo, K. R., & Figueiredo, J. C. (2020). *Building a web-based recommendation system for restaurants using Flask*. In the International Conference on Computational Processing of the Portuguese Language (pp. 310-319).

V. Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). *Item-based collaborative filtering recommendation algorithms*. In Proceedings of the 10th International Conference on World Wide Web (pp. 285-295).

VI. Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E. P., Yan, H., & Li, X. (2018). *Modeling the Assimilation-Contrast Effects in Online Product Rating Systems: Debiasing and Recommendations*. ACM Transactions on Information Systems (TOIS), 36(1), 1-37.

VII. Zhang, S., Zhao, H., Liu, J., Yang, X., Zhang, Z., & Zhai, C. (2019). *Improving top-n recommendation via parallel diversification.* In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1286-1296).