# REPORT WORK

**Abstract**—In this report, we present the design and implementation of a cuisine recommendation system aimed at assisting users in discovering recipes that align with their tastes and dietary requirements. Leveraging machine learning techniques such as content-based and collaborative filtering, our system analyzes user preferences, ingredient profiles, and historical interactions to generate personalized recipe suggestions.

**PROBLEM BEING ADDRESSED-** The problem being addressed in this project is the need for a robust cuisine recommendation system that can assist users in discovering recipes aligned with their tastes, preferences, and dietary requirements. With the vast amount of recipe data available online, users often struggle to find recipes that suit their specific needs and interests. Additionally, users may also desire personalized recommendations based on their past interactions and feedback.

**RELEVANT LITERATURE**- Existing research in cuisine recommendation systems has focused on collaborative filtering, content-based filtering, and hybrid approaches. However, these methods face challenges such as sparsity of user-item interaction data, limited diversity in recommendations, and scalability issues. The cold-start problem remains a key challenge, especially for new users or items with sparse data. Future research should aim to address these limitations by developing more effective recommendation techniques, enhancing diversity in recommendations, and improving scalability and efficiency. Also, there is no significant way to evaluate the accuracy of the recommendations. We have tried to do so by using a cosine similarity heatmap and precision at k. Additionally, advancements in deep learning, natural language processing, and user modeling can offer new opportunities for enhancing the accuracy, diversity, and personalization of recipe recommendations.

**DATASET-** For our project, we utilize a comprehensive dataset from Kaggle, comprising a diverse collection of Food recipes showcasing various regional cuisines and culinary traditions. The dataset includes essential information such as recipe names, ingredients, preparation time, nutritional content, and user ratings. In the earlier deadline we stated that we will be using. https://www.kaggle.com/datasets/kritirathi/indian-fooddataset-with/data but due to less rows we have changed our dataset to https://www.kaggle.com/datasets/irkaal/foodcomrecipes-and-reviews.

**DATA PREPROCESSING** -In this section, we detail the preprocessing steps applied to the recipe dataset to prepare it for further analysis and modeling. A. Reading and Filtering Dataset The recipe dataset is initially read using the Pandas library, loading it into a DataFrame named data. Subsequently, irrelevant columns are filtered out to focus on essential attributes relevant to the recommendation system to reduce noise prevent unnecessary complexity. B. Removing Unnecessary Rows and Columns To enhance data consistency, we removed entries with missing ratings, incorrect/erroneous data etc. using the drop function. C. Selecting Diverse Rating Dishes Further, we stratified the dataset based on aggregated ratings, selecting a balanced mix of highly rated and lower-rated recipes for model training. D. Modifying Column Formats, Handling Missing and Erroneous Data Additionally, we brought the entries of date of publication, cooking time, ingredients columns into the required format for efficient handling of data. We also handled missing values in various columns, ensuring consistency and completeness. E. Removing Duplicate Entries Duplicate recipe names are removed from the dataset, ensuring uniqueness and avoiding redundancy in recommendations. F. Merging Datasets The preprocessed recipe dataset is merged with the reviews dataset based on RecipeId, facilitating the integration of user ratings into the recommendation system. G. Filtering Reviews Reviews from reviewers with less than five reviews are removed from the dataset, ensuring that only substantial reviews are considered in the analysis. H. Saving Processed Data The final preprocessed dataset is saved to a new file for further analysis and modeling.

**DATA VISUALIZATION**- In this section, we present visualizations of various aspects of the processed dataset to gain insights into recipe categories, rating distribution, publication trends, calorie distribution, and top recipes and authors based on review count. A. Distribution of Recipe Categories The distribution of recipe categories is visualized using a pie chart. Rare categories, representing less than 1 percent of the dataset, are grouped under ’Others’ to simplify to simplify the visualization.

**CONTENT-BASED FILTERING**-

Content-based filtering is a recommendation technique that suggests items to users based on their preferences and item features. In this section, we describe the implementation of content-based filtering for recipe recommendations using ingredient information. A. Extracting Unique Recipes First, unique recipes are extracted from the dataset to ensure that each recipe is considered only once for recommendation. B. TF-IDF Vectorization TF-IDF (Term Frequency-Inverse Document Frequency) vectorization is applied to the recipe ingredient parts to convert text data into numerical vectors. This process captures the importance of each ingredient in the recipe while considering its frequency across all recipes. C. Computing Cosine Similarity and Indexing Cosine similarity is calculated between the TF-IDF vectors of recipes to measure the similarity between them. An index is created to map recipe names to their corresponding indices in the dataset for efficient retrieval. D. Recommendation Function A recommendation function is defined to provide recommendations based on the similarity scores calculated using cosine similarity. Given a recipe title, the function returns a list of top recommended recipes with similar ingredient profiles. E. Printing Recommendations Recommendations are printed for sample recipe titles, such as ’Buttermilk Pie’ and ’Potato Salad’. The function returns a list of recommended recipes based on the ingredient similarities with the input recipe. — Example Recommendations: For the input recipe ’Buttermilk Pie’, the following recommendations are provided: • Chocolate Dessert Crepes • Easy Red Velvet Cake • Red Velvet Waffles • Filled Coffee Cake • Peanut Butter Fudge Cake For the input recipe ’Potato Salad’, the following recommendations are provided: • Amish Potato Salad • Mom’s Danish Potato Salad • Kathy’s Macaroni Salad • Curry Deviled Eggs With Cilantro • Pickled Eggs F. Calculation of Diversity Ratio The diversity ratio is calculated as the ratio of the number of unique recommended items to the total number of recommendations made. This metric provides insights into the variety and diversity of recommendations generated by the contentbased filtering model.

COLLABORATIVE FILTERING -Collaborative filtering is a recommendation technique that identifies patterns in user behavior and preferences to recommend items to users. In this section, we describe the implementation of collaborative filtering for recipe recommendations using a reviewer-recipe-rating matrix. A. Creating Reviewer-Recipe-Rating Matrix A reviewer-recipe-rating matrix is constructed from the processed dataset, where each row represents a reviewer, each column represents a recipe, and the values represent the ratings given by reviewers to recipes. Missing ratings are filled with zeros. Ratings may also be normalised to penalise ratings of those who ovverrate/underrate irratically. B. Calculating Cosine Similarity Matrix Cosine similarity is computed between reviewers based on their rating vectors using the reviewer-recipe-rating matrix. This similarity matrix captures the similarity between pairs of reviewers, indicating how alike their preferences are. C. Predicting Ratings for Missing Entries Predictions for missing ratings for each user are made one by one. This involves identifying similar users and using their ratings for the target recipe to predict missing rating. The predictions are based on the cosine similarity scores between users. It is used to in a way to assign optimum weights to each other reviewer's rating. The users having rating vector similar to target user will have higher weight and vice versa. Weighted sum of these ratings is taken to predict target user's rating for the recipe.