CUISINE RECOMMENDATION SYSTEM

Shyama Goel (2021492)
Shivam (2021489)

INDRAPRASTHA INSTITUTE *of* INFORMATION TECHNOLOGY **DELHI**

CUISINE RECOMMENDATION

With an increasing abundance of online recipe data, users often face challenges in finding recipes that align with their unique tastes, preferences, and dietary requirements.

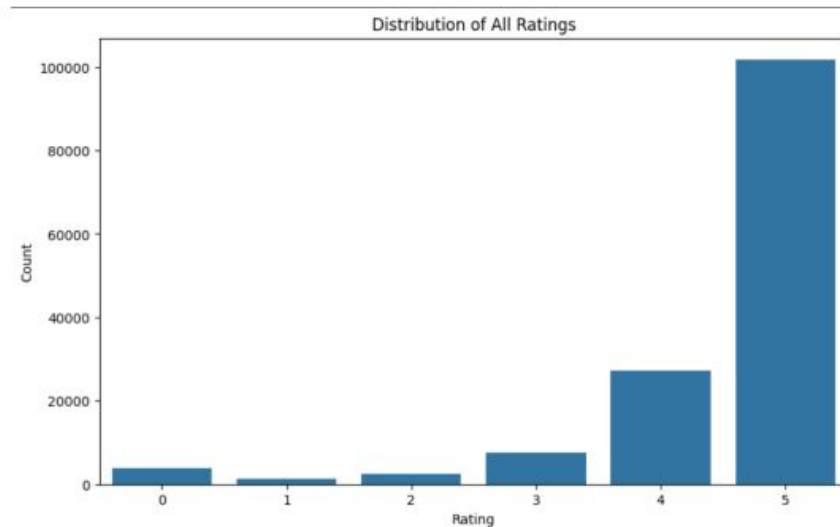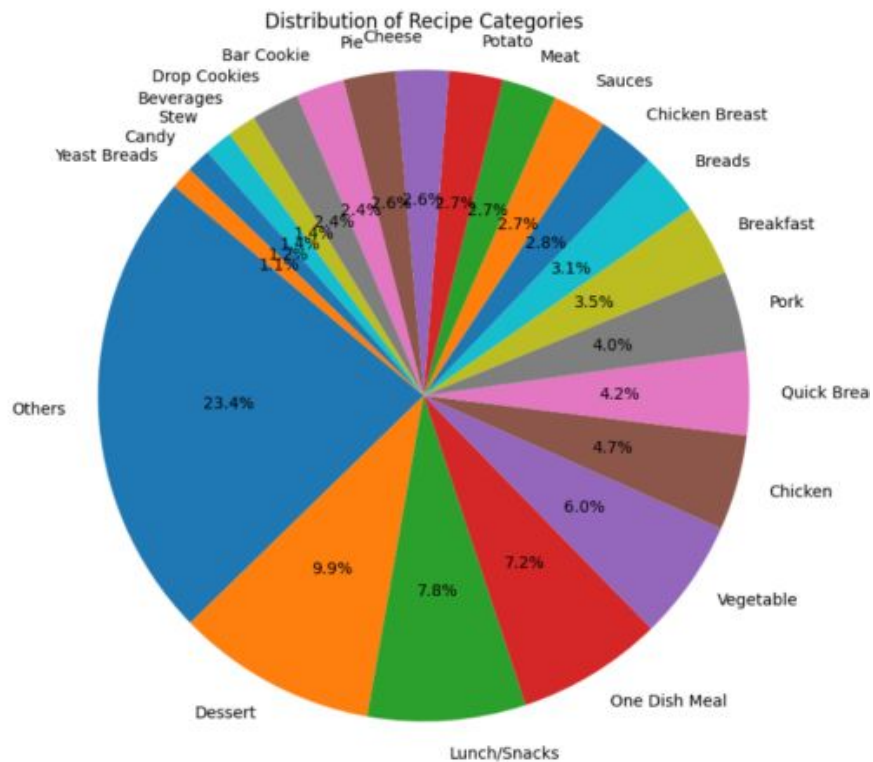**IIID** | INDRAPRASTHA INSTITUTE *of* INFORMATION TECHNOLOGY **DELHI**

Dataset used (Source-kaggle)

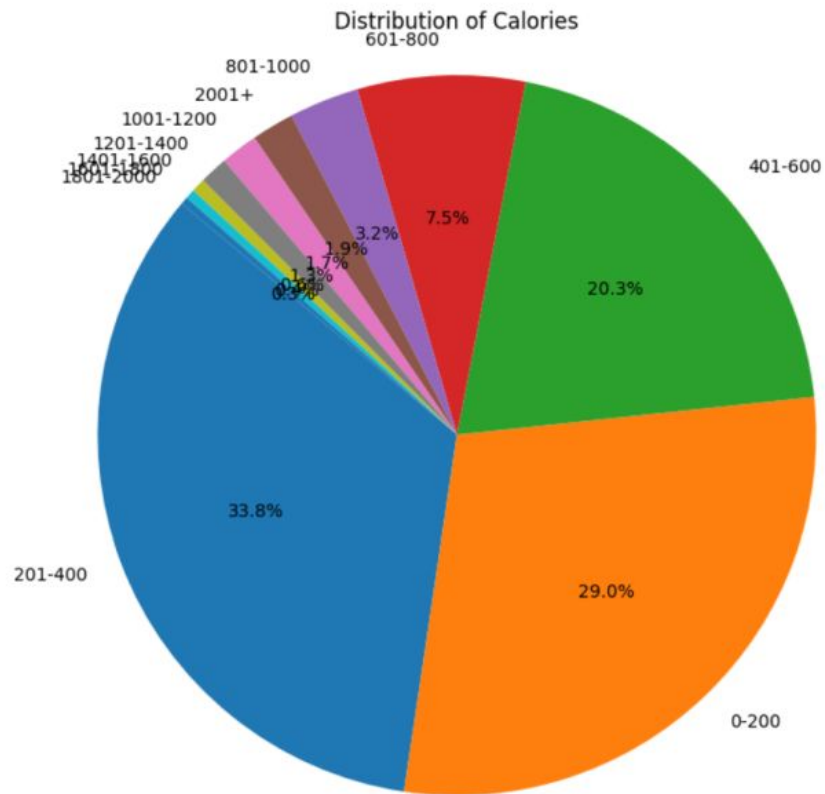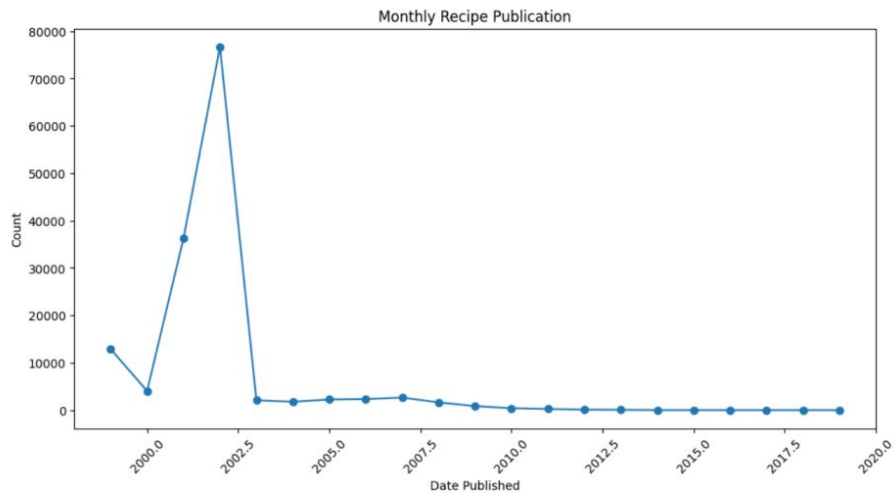| RecipeId	Name	AuthorId	AuthorName	CookTime	PrepTime	DatePublished	RecipeCategory	AggregateRating	ReviewCount	RecipeIngredient	Calories	FatContent	SaturatedFat	Cholesterol	SodiumContent	Carbohydrate	FiberContent	SugarContent
54	Carrot Cak	1535	Marg Cayr	50	45	1999	Dessert	5	17	c("carrots"	522.6	27.1	4.9	69.8	534.8	67	1.6	47.9
56	Buttermilk	1581	thefensk	60	20	1999	Pie	5	171	c("butter"	395.9	19.1	9.8	101.5	269.8	52.2	0.7	39.9
56	Buttermilk	1581	thefensk	60	20	1999	Pie	5	171	c("butter"	395.9	19.1	9.8	101.5	269.8	52.2	0.7	39.9
72	Brownie P	193853	MirandaLe	35	40	1999	Dessert	5	10	c("all-purp	182.8	5.9	3.6	7.8	160.2	34	2.1	25.3
76	Alfredo Sa	1535	Marg Cayr	10	5	1999	Sauces	5	207	c("sweet b	489.9	50.4	31.5	159	312.4	2.5	0	0.2
76	Alfredo Sa	1535	Marg Cayr	10	5	1999	Sauces	5	207	c("sweet b	489.9	50.4	31.5	159	312.4	2.5	0	0.2
76	Alfredo Sa	1535	Marg Cayr	10	5	1999	Sauces	5	207	c("sweet b	489.9	50.4	31.5	159	312.4	2.5	0	0.2
76	Alfredo Sa	1535	Marg Cayr	10	5	1999	Sauces	5	207	c("sweet b	489.9	50.4	31.5	159	312.4	2.5	0	0.2
92	Brown Bag	1533	Dancer	0	10	1999	Oranges	5	6	c("oranges	141.4	5	0.5	0	1.5	25.4	3.3	20.6
93	Blue Stuffe	1533	Dancer	0	15	1999	Free Of...	5	11	c("blue ch	246.3	22.2	12.1	53.8	482.8	3.5	0.9	1
129	Champagn	1592	polar-expr	0	5	1999	Punch Bev	5	30	c("frozen l	236.8	0.2	0	0	14.2	41.6	0.2	38.1
135	Cherry Sar	1593	Traci Evan	0	30	1999	Lunch/Sna	5	5	c("powder	73.3	0.8	0.2	0	171.3	14.3	0.6	1.1
143	Cherry Tor	2178	troyh	70	35	1999	Weeknigh	5	7	c("cherry t	488.6	38.6	10	24.8	483.5	23.7	2.9	7.2
143	Cherry Tor	2178	troyh	70	35	1999	Weeknigh	5	7	c("cherry t	488.6	38.6	10	24.8	483.5	23.7	2.9	7.2
150	All-Purpos	1536	Hilary	520	20	1999	Whole Chi	5	12	c("chicken	247.2	17.3	5	86.2	99.9	0	0	0
153	Amish Frie	1540	gowiththe	70	0	1999	Quick Brea	5	50	c("sugar",	615.9	21.3	4	42.4	374.9	100	2.3	65.2

Final Processed Dataset (1) (1)

Number of Rows: 16,691
Number of Columns: 25
Our dataset contains 16,691 rows and 25 columns, providing detailed information about recipes, including their names, categories, cooking/preparation times, ingredients, and nutritional content (e.g., calories, fat, sugar, protein). It also features user-generated data like aggregated ratings, review counts, and individual reviewer feedback, along with metadata about recipe authors.

Data Visualizations

Distribution of Recipe Categories

Distribution of All Ratings

Monthly Recipe Publication

Distribution of Calories

CONTENT-BASED FILTERING

Unique Recipe Extraction: Ensure only unique recipes are used for recommendations.

TF-IDF Vectorization: Convert recipe ingredients into numerical vectors based on their importance and frequency across recipes.

Cosine Similarity: Measure similarity between recipes by comparing their TF-IDF vectors.

Recommendation Function: Suggests recipes with similar ingredients for a given recipe.

Example Recommendations:

- **Input**: *Buttermilk Pie* → Suggested recipes include *Chocolate Dessert Crepes* and *Red Velvet Waffles*.
- **Input**: *Potato Salad* → Suggested recipes include *Amish Potato Salad* and *Curry Deviled Eggs With Cilantro*.

Printing Recommendations:

```python
print(get_recommendations('Brownie Pudding'))
print(get_recommendations('Brown Bag Apple Salad'))
print(get_recommendations('Chicken Curry'))
```

```
['Chocolate Pudding Cake', 'Apricot Squares', 'Chocolate Doughnuts', 'Hot Fudge Pudding Cake II', 'Fresh Peach Upside Down Cake']
['Chocolate Coated Orange Peels', 'Yankee Grapefruit Meringue Pie', 'Summer Fruit Bowl', 'Hemingway Special- a Caribbean Inspired Cocktail', 'Chick
['Chicken Biryani II', 'Exotic and Sweet Moroccan Chicken with Spicy Rice', 'Easy African Sweet Potato Patties', 'Indian Corn Pilaf', 'Basmati Rice
```

BENEFITS AND LIMITATIONS

Personalized Recommendations:

- Based on the input recipe's ingredients, users get recommendations tailored to their preferences.

Efficient Retrieval:

- Pre-computed similarity matrix enables quick recommendations.

Cold-Start Problem:

- Recipes with missing or limited ingredient data may not yield accurate recommendations.

Ingredient Ambiguity:

- Ingredient variations (e.g., "chicken breast" vs. "chicken") might impact similarity calculations.

Calculation of Diversity Ratio:

```python
recommendations = {}
for title in unique_df['Name']:
    recs = get_recommendations(title)
    recommendations[title] = recs

num_unique_items = len(set(item for sublist in recommendations.values() for item in sublist))
num_recommendations = sum(len(recs) for recs in recommendations.values())
diversity_ratio = num_unique_items / num_recommendations

print("Diversity Ratio:", diversity_ratio)
```

```
Diversity Ratio: 0.17871140939597316
```

Diversity Ratio=Total Number of Recommendations Number of Unique Recommended Items

A **higher diversity ratio** indicates that the recommendations are more varied (less repetitive across recipe titles).

Heat Map

Cosine Similarity Heatmap after filtering
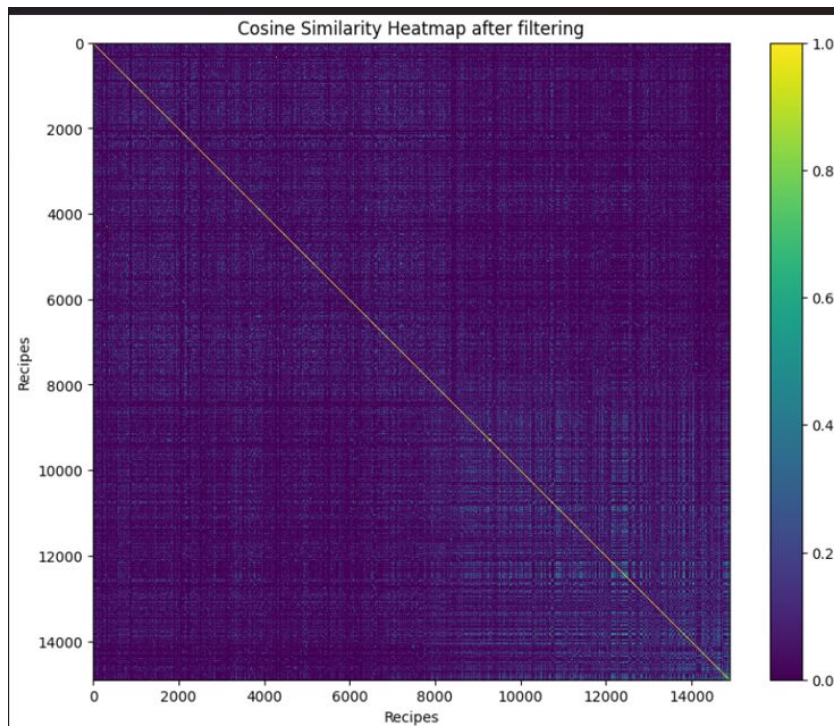
COLLABORATIVE FILTERING-Collaborative filtering is a recommendation technique that

relies on **user behavior** (ratings) to predict preferences. It assumes that users with similar preferences will rate items similarly

Reviewer-Recipe-Rating Matrix:

- A matrix is created where **rows represent reviewers**, **columns represent recipes**, and **values represent ratings** given by reviewers.
- Missing ratings are filled with zeros.

ReviewerName	Recipe 1	Recipe 2	Recipe 3
Reviewer 1	5	0	0
Reviewer 2	0	4	0
Reviewer 3	0	0	3

Cosine Similarity Matrix:

- Calculates **cosine similarity** between reviewers based on their rating patterns.
- The similarity score (e.g., 1.0 for identical, 0.8 for somewhat similar) indicates how similar two reviewers' preferences are.

Predicting Missing Ratings:

- For a missing rating, the algorithm:
  - **Finds similar reviewers** using the similarity matrix.
  - Assigns higher weights to ratings from reviewers with higher similarity scores.
  - Calculates the **weighted average** of these ratings to predict the missing rating.
- This allows predicting a user's rating for a recipe based on preferences of similar users