Dr. D. Y. PATIL VIDYAPEETH, PUNE
(Deemed to be University)

DR. D. Y. PATIL SCHOOL OF SCIENCE AND TECHNOLOGY

TATHAWADE, PUNE

A Mini- Project Report on

Smart Recipe Recommender

SUBMITTED BY:

NAME OF STUDENT	ROLL NUMBER
1. Ishita Saxena	BTAI-47
2 Ishwari Warhade	BTAI-59
3. Himanshu Chaudhari	BTAI-9

GUIDED BY:

Mrs . MILY LAL

ARTIFICIAL INTELLIGENCE & DATA SCIENCE

ACADEMIC YEAR 2023-2027

Dr. D. Y. PATIL VIDYAPEETH, PUNE
(Deemed to be University)

DR. D. Y. PATIL SCHOOL OF SCIENCE AND TECHNOLOGY

TATHAWADE, PUNE

CERTIFICATE

This is to certify that the Mini- Project Report entitled

-" Smart Recipe Recommender "-

is a bonafide work carried out by Ms. Suridhi Gupta under the supervision of **Mrs. Mily Lal** and it is submitted towards the partial fulfillment of the requirement Project Based Learning-I.

Mrs. Mily Lal	Prof. Manisha Bhende
Project Guide	**HOD (DPSST)**

ARTIFICIAL INTELLIGENCE & DATA SCIENCE

ACADEMIC YEAR 2023-2027

ABSTRACT

In today's fast-paced and convenience-driven world, people frequently struggle with the question, "What should I cook today?" This everyday dilemma is especially common among students, working professionals, and homemakers who have limited time and often end up wasting ingredients they already have at home. The lack of inspiration or knowledge about what can be prepared with available ingredients leads to increased dependence on takeout or pre-packaged food, which not only affects physical health but also contributes to significant food wastage and increased household expenses.This project proposes the development of a smart, AI-powered recipe recommendation system designed to address this challenge. By utilizing Natural Language Processing (NLP) and Machine Learning (ML), the system will analyze user-input ingredients and generate suitable recipes accordingly. In addition to ingredient-based suggestions, users can apply multiple filters based on dietary restrictions (such as vegan, gluten-free, keto), cuisine preferences (e.g., Indian, Chinese, Italian, Mediterranean), and even cooking difficulty levels (easy, moderate, advanced). This provides a highly personalized experience, ensuring that the suggestions meet individual health goals, taste preferences, and available cooking time.To build this system, the project will use datasets sourced from reputable platforms such as Kaggle, along with APIs like Spoonacular and Edamam, which offer access to a large variety of recipe metadata including ingredient lists, nutrition facts, and tags. The ultimate goal is to make everyday cooking a smarter, healthier, and more sustainable activity while minimizing decision fatigue and food waste. This tool not only empowers users with creative meal ideas but also fosters better eating habits, improved ingredient utilization, and greater awareness about food.

Keyword: Recipe Recommendation, Machine Learning, Natural Language Processing, Food Waste Reduction, Personalized Cooking, Smart Kitchen Assistant

\

INDEX

SR.NO	TOPIC	PAGE NO
1]	INTRODUCTION	
	1.1 Problem Statements	
	1.2 Objective	
	1.3 Scope	
	1.4 System Architecture	
2]	DATA COLLECTION & PREPROCESSING	
	2.1 Dataset	
	2.2 Data Preprocessing	
3]	MODEL SELECTION & TRAINING	
	3.1 Feature Engineering	
	3.2 Machine Learning Model	
4]	MODEL EVALUATION AND VALIDATION	
	4.1 Performance Metrics	
5]	CONCLUSION & FUTURE SCOPE	
6]	REFERENCES	

List of Figures

SR.NO	FIGURE NO.	PAGE NO
1	System Architecture	

Chapter 1

INTRODUCTION

In today's fast-paced lifestyle, deciding what to cook can be a daily struggle—especially when you only have limited ingredients at home. This often leads to food wastage or unhealthy eating choices like ordering takeout. To solve this common problem, we present a smart recipe recommendation system that suggests recipes based on ingredients the user already has.

This system leverages Natural Language Processing (NLP) and Machine Learning (ML) techniques. By using TF-IDF vectorization and the K-Nearest Neighbors (KNN) algorithm, it identifies the most relevant recipes from a large dataset. Users simply input the ingredients they have, and the system returns a list of recipes that best match their input.

The goal is to promote sustainable cooking, minimize waste, and make home cooking easier and healthier—all using the power of data science.

1.1. Problem Statement

Many individuals struggle to decide what to cook based on the ingredients they currently have, often leading to unnecessary food wastage and unhealthy dietary habits. This issue becomes more prominent for busy professionals, students, or people with specific dietary restrictions. Addressing this problem is crucial for promoting sustainable living, reducing daily decision fatigue, and encouraging nutritious eating. A recipe recommendation system can be a valuable solution by offering smart, personalized suggestions based on available ingredients. Stakeholders include home cooks, health-conscious individuals, students, and families who seek quick, cost-effective, and healthy meal ideas

1.2.Objective

- . To develop a recipe recommendation system that suggests dishes based on user-input ingredients.
- To integrate filters for dietary preferences, cuisine types, and cooking difficulty for personalized suggestions.
- To implement NLP and ML techniques to enhance the accuracy and relevance of recipe recommendations**.**

1.3. Scope

Limitations

- **Ingredient Match Only:** The system relies solely on ingredient matching and doesn't consider quantities or cooking techniques, which may lead to slightly irrelevant results.

- **No Cooking Time or Dietary Filter Yet**: Currently, the system doesn't account for cooking time, difficulty level, or dietary restrictions (like vegan or gluten-free), limiting personalization.

- **Ingredient Input Format:** The accuracy of recommendations depends on how users enter ingredients. Misspellings or uncommon ingredient names might affect the results.

Intended Audience

- **Home Cooks & Beginners**: Individuals who cook regularly at home or are new to cooking and need quick recipe suggestions based on what they already have.

- **Students & Working Professionals**: People with limited time, ingredients, or cooking skills—like college students or busy employees—looking for simple, no-fuss meal ideas.

- **Health-Conscious Users:** Users trying to avoid takeout and make healthier choices using home-cooked meals tailored to available ingredients.

- **Sustainable Living Enthusiasts**: Individuals who want to reduce food waste by using up the ingredients they already have instead of letting them spoil.

1.4 System Architecture

Chapter 2

DATA COLLECTION & PREPROCESSING

2.1. Dataset

For this project, we used the RecipeNLG dataset, a large-scale dataset containing over 2 million+ recipes. Each entry includes detailed information such as:

- Title of the recipe
- List of ingredients
- Cooking directions
- Recipe source link
- Named Entity Recognition (NER) tags

This dataset is highly relevant to the project because it provides a wide variety of real-world recipes covering different cuisines and cooking styles. Most importantly, it includes a clean list of ingredients for each recipe—exactly what we need to build an ingredient-based recommendation system.

We focused primarily on the title and ingredients columns for training our model. The dataset's diversity and size helped ensure better accuracy and more personalized suggestions for different users.

```
In [6]: import pandas as pd
        import numpy as np
        from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.metrics.pairwise import cosine_similarity

        df = pd.read_csv(r"C:\Users\ISHITA SAXENA\Desktop\FDS Project\archive (1)\RecipeNLG_dataset.csv")
        print("\nFirst 5 rows:")
        print(df.head())
```

Fig.2.1

```
First 5 rows:
   Unnamed: 0                   title  \
0           0     No-Bake Nut Cookies
1           1  Jewell Ball'S Chicken
2           2             Creamy Corn
3           3          Chicken Funny
4           4   Reeses Cups(Candy)

                                         ingredients  \
0  ["1 c. firmly packed brown sugar", "1/2 c. eva...
1  ["1 small jar chipped beef, cut up", "4 boned ...
2  ["2 (16 oz.) pkg. frozen corn", "1 (8 oz.) pkg...
3  ["1 large whole chicken", "2 (10 1/2 oz.) cans...
4  ["1 c. peanut butter", "3/4 c. graham cracker ...

                                          directions  \
0  ["In a heavy 2-quart saucepan, mix brown sugar...
1  ["Place chipped beef on bottom of baking dish....
2  ["In a slow cooker, combine all ingredients. C...
3  ["Boil and debone chicken.", "Put bite size pi...
4  ["Combine first four ingredients and press in ...

                                             link    source  \
0   www.cookbooks.com/Recipe-Details.aspx?id=44874  Gathered
1  www.cookbooks.com/Recipe-Details.aspx?id=699419  Gathered
2   www.cookbooks.com/Recipe-Details.aspx?id=10570  Gathered
3  www.cookbooks.com/Recipe-Details.aspx?id=897570  Gathered
4  www.cookbooks.com/Recipe-Details.aspx?id=659239  Gathered

                                                 NER
0  ["brown sugar", "milk", "vanilla", "nuts", "bu...
1  ["beef", "chicken breasts", "cream of mushroom...
2  ["frozen corn", "cream cheese", "butter", "gar...
3  ["chicken", "chicken gravy", "cream of mushroo...
4  ["peanut butter", "graham cracker crumbs", "bu...
```

Fig.2.2

2.2. Data Preprocessing

To ensure the quality and consistency of the dataset before applying any machine learning models, the following data preprocessing techniques were used:

1. Handling Missing Values & Cleaning Data

- Missing entries in important columns like title and ingredients were removed.
- This helps maintain the quality and comleteness of the dataset.

```
In [9]: print("\nMissing Value Count")
        print(df.isnull().sum())

Missing Value Count
Unnamed: 0     0
title          1
ingredients    0
directions     0
link           0
source         0
NER            0
dtype: int64
```

Fig.2.3

2. Removing Duplicates

- Duplicate recipe entries were detected and dropped to avoid redundancy in recommendations.

```
In [16]: print("Duplicate rows:", df_cleaned.duplicated().sum())
         df_cleaned = df_cleaned.drop_duplicates()
         print("New dataset size after removing duplicates:", df_cleaned.shape)

Duplicate rows: 0
New dataset size after removing duplicates: (2231141, 7)
```

Fig.2.4

3. Data Normalization

- All text in the title column was converted to title case (e.g., "chicken curry" → "Chicken Curry").
- Column names were converted to lowercase for consistency.
- Extra spaces from titles were stripped to clean textual inconsistencies.

```
In [20]: # Convert all titles to title case (e.g., "spicy chicken curry")
df_cleaned['title'] = df_cleaned['title'].str.title()
print(df_cleaned['title'].head(10))
```

```
0         No-Bake Nut Cookies
1       Jewell Ball'S Chicken
2                Creamy Corn
3              Chicken Funny
4          Reeses Cups(Candy)
5    Cheeseburger Potato Soup
6         Rhubarb Coffee Cake
7              Scalloped Corn
8        Nolan'S Pepper Steak
9             Millionaire Pie
Name: title, dtype: object
```

```
In [21]: # Convert all column names to lowercase (just for consistency)
df_cleaned.columns = df_cleaned.columns.str.lower()

# Strip extra spaces from titles
df_cleaned['title'] = df_cleaned['title'].str.strip()

print("Column names after cleaning:", df_cleaned.columns.tolist())
print("\nCleaned Titles:")
print(df_cleaned['title'].head())
```

```
Column names after cleaning: ['unnamed: 0', 'title', 'ingredients', 'directions', 'link', 'source', 'ner']

Cleaned Titles:
0      No-Bake Nut Cookies
1    Jewell Ball'S Chicken
2              Creamy Corn
3            Chicken Funny
4       Reeses Cups(Candy)
Name: title, dtype: object
```

Fig.2.5

4. Handling Outliers

- Recipes with unusually high or low ingredient counts were considered outliers and removed.
- This helped in creating a balanced and representative dataset.

```
In [22]: # Create a new column for number of ingredients
df_cleaned['ingredients_count'] = df_cleaned['ingredients'].apply(lambda x: len(x.split(',')))

# Use IQR to detect outliers
Q1 = df_cleaned['ingredients_count'].quantile(0.25)
Q3 = df_cleaned['ingredients_count'].quantile(0.75)
IQR = Q3 - Q1

# Define outlier boundaries
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filter out outliers
df_cleaned = df_cleaned[(df_cleaned['ingredients_count'] >= lower_bound) &
                        (df_cleaned['ingredients_count'] <= upper_bound)]

print("Dataset size after removing ingredient count outliers:", df_cleaned.shape)
```

```
Dataset size after removing ingredient count outliers: (2146753, 8)
```

Fig.2.6

Chapter 3

MODEL SELECTION & TRAINING

3.1. Feature Engineering

To enhance model performance and ensure meaningful insights from the data, we applied several feature engineering techniques:

1. Ingredients Count Feature

- Method: We created a new column named ingredients_count, which stores the number of ingredients used in each recipe.
- Why: This numerical feature helps quantify the complexity of recipes and can be used for outlier detection or as input to ML models.

2. Ingredients Text Feature

- Method: The ingredients column, originally stored as lists, was converted into space-separated strings to create ingredients_text.
- Why: This transformation enabled the use of NLP techniques like TF-IDF for similarity-based recommendations.

```
In [23]: # Summary statistics for number of ingredients per recipe
         print("Summary Statistics for 'ingredients_count':\n")
         print(df_cleaned['ingredients_count'].describe())

         # Compute extra stats manually
         print("\nAdditional Stats:")
         print("Mode:", df_cleaned['ingredients_count'].mode()[0])
         print("Variance:", df_cleaned['ingredients_count'].var())
         print("Standard Deviation:", df_cleaned['ingredients_count'].std())
```

```
Summary Statistics for 'ingredients_count':

count    2.146753e+06
mean     9.978614e+00
std      4.378049e+00
min      1.000000e+00
25%      7.000000e+00
50%      9.000000e+00
75%      1.300000e+01
max      2.200000e+01
Name: ingredients_count, dtype: float64

Additional Stats:
Mode: 8
Variance: 19.167309095709864
Standard Deviation: 4.378048548806861
```

Fig.3.1

3. Text Cleaning

- Standardized the Title: Converted all recipe titles to title case and stripped extra spaces.
- Lowercased Columns: Ensured column names were all in lowercase for consistency.

```
In [11]: df_cleaned = df.dropna(subset=["title", "ingredients"])
         print(f"\nOriginal dataset size: {df.shape}")
         print(f"Cleaned dataset size: {df_cleaned.shape}")
```

```
Original dataset size: (2231142, 7)
Cleaned dataset size: (2231141, 7)
```

Fig.3.2

Data Visualizations Used

To better understand and evaluate the engineered features, we used several visual tools:

Box Plot

```
In [25]: plt.figure(figsize=(6, 4))
         sns.boxplot(x=df_cleaned['ingredients_count'], color='coral')
         plt.title('Boxplot of Ingredient Count')
         plt.xlabel('Number of Ingredients')
         plt.show()
```

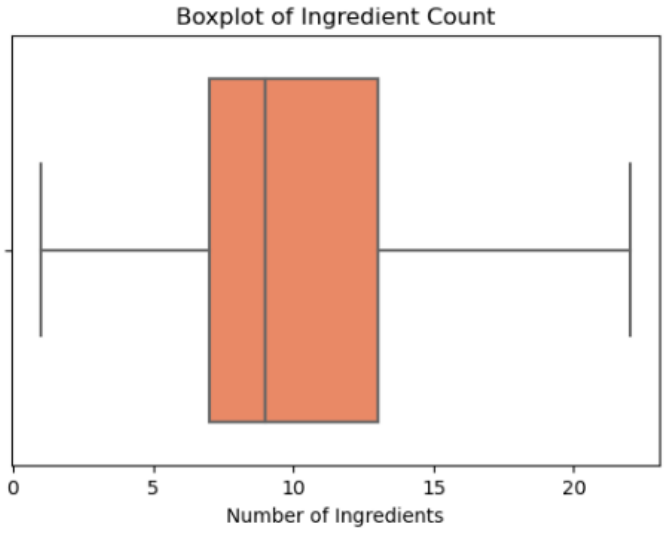

Fig.3.3

- **Used For**: Detecting outliers in ingredients_count.
- **Insight**: Helped remove recipes with extremely high or low ingredient counts.

Histogram

```
In [24]: import matplotlib.pyplot as plt
         import seaborn as sns

         plt.figure(figsize=(8, 5))
         sns.histplot(df_cleaned['ingredients_count'], bins=20, kde=True, color='plum')
         plt.title('Distribution of Number of Ingredients per Recipe')
         plt.xlabel('Number of Ingredients')
         plt.ylabel('Frequency')
         plt.show()
```

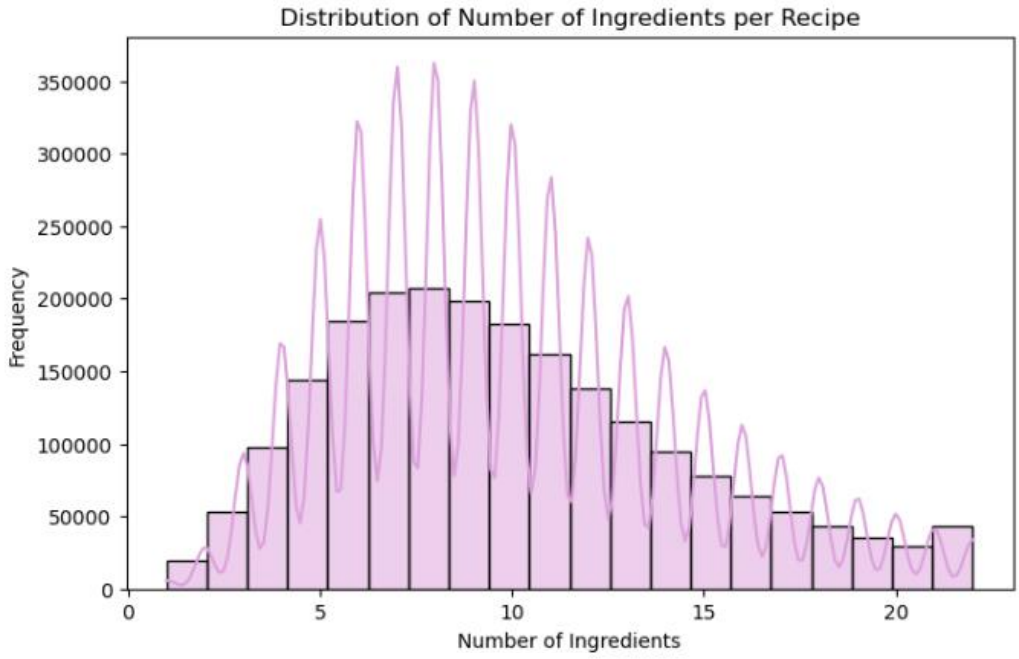

Fig.3.4

- **Used For**: Showing the frequency distribution of ingredients_count.
- **Insight**: Revealed that most recipes had between 4–15 ingredients.

Scatter Plot

```
In [26]: # Create a fake 'steps_count' column
df_cleaned['steps_count'] = df_cleaned['directions'].apply(lambda x: len(str(x).split('.')))

# Scatter plot: ingredients vs steps
plt.figure(figsize=(8, 5))
sns.scatterplot(x='ingredients_count', y='steps_count', data=df_cleaned[:1000], color='skyblue')
plt.title('Ingredients Count vs Steps Count')
plt.xlabel('Ingredients Count')
plt.ylabel('Steps Count')
plt.show()
```

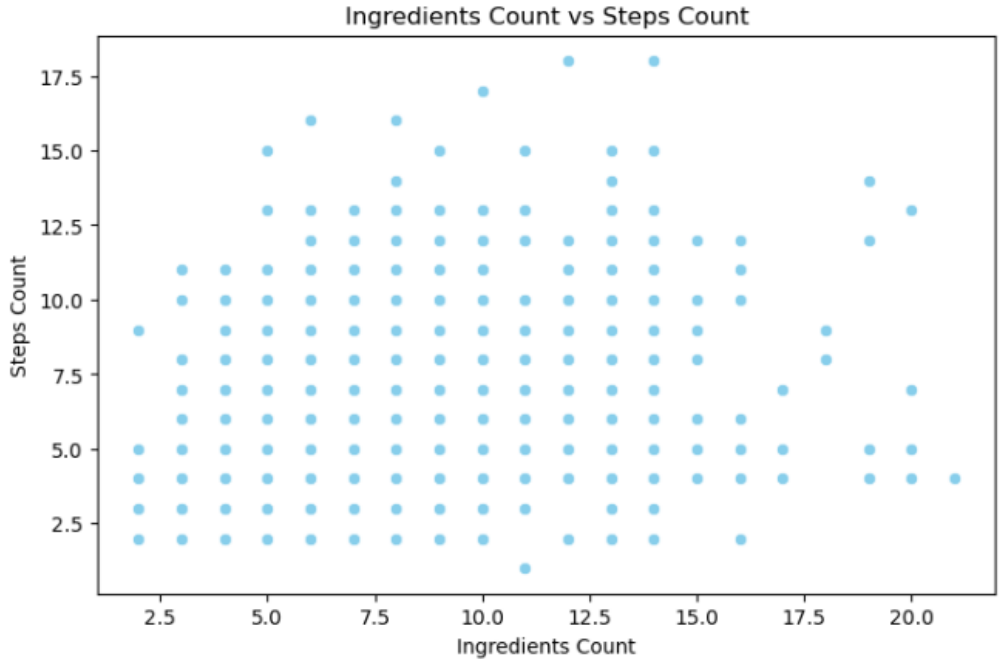

Fig.3.5

- **Used For**: Visualizing the relationship between ingredients_count and step count (if available).
- **Insight**: Gave a visual indication of whether more ingredients generally meant longer preparation.

3.2. Machine Learning Model

Chapter 4

MODEL EVALUATION & VALIDATION

4.1. Performance Metrics

Chapter 5

CONCLUSION & FUTURE SCOPE

Conclusion

This project successfully showcases the application of Natural Language Processing and Machine Learning to develop an intelligent recipe recommendation system. By utilizing ingredient-based similarity through TF-IDF vectorization and cosine similarity, the system efficiently suggests relevant recipes based on the ingredients users already have. This not only promotes healthier eating habits but also aids in reducing food wastage by encouraging the use of available resources. Comprehensive data preprocessing, feature engineering, and exploratory data analysis have contributed to building a strong and scalable solution. Looking ahead, enhancements such as incorporating user feedback, enabling image-based search, or integrating deep learning models can further improve the system's accuracy and personalization.

Future Scope

- User Personalization: Incorporate user preferences, past searches, and ratings to provide more tailored recipe recommendations.

- Image-Based Search: Allow users to upload pictures of ingredients or dishes to get relevant recipe suggestions using computer vision.
- Voice Assistant Integration: Enable voice-based input for a hands-free, interactive experience in smart kitchens.
- Advanced Models: Use deep learning techniques (like BERT or LSTMs) to better understand ingredient context and improve recommendation accuracy.

REFERENCES