Text Mining Project By Krishaang Anand

1. Introduction

This project focuses on applying text mining and natural language processing (NLP) techniques to analyse traditional Uttarakhand dish recipes. The aim was to explore how unstructured text (ingredients and recipes) can be converted into meaningful insights, patterns, and intelligent recommendations. Instead of treating recipes as plain text, this project transforms them into structured data that can be analysed using machine learning techniques to understand ingredient importance, dish similarity, and cuisine patterns.

This project moves beyond simple keyword searching and demonstrates how data mining concepts can be applied to cultural and culinary datasets, making it both academically relevant and practically useful.

2. Objective of the Project

The main objectives of this text mining project were: - To analyse ingredient patterns in traditional Uttarakhand dishes. - To identify the most frequently used ingredients. - To calculate ingredient importance using TF-IDF. - To group similar dishes using clustering. - To recommend similar dishes based on mathematical similarity. - To visualise text data using graphs and word clouds.

Overall, the goal was to convert recipe text into actionable insights using text mining techniques.

3. Dataset Description

The dataset used in this project consists of traditional Uttarakhand recipes stored in a CSV file. The dataset contains the following main columns:

- Dish_Name: Name of the dish
- Ingredients: Comma-separated list of ingredients
- Recipe: Instructions for preparation

Each row represents a single recipe. The dataset was manually curated to preserve cultural authenticity but contains limitations which will be discussed later.

4. Tools and Libraries Used

4.1 Pandas

Used to load, clean and manipulate the dataset. It allows structured handling of CSV data and easy column operations.

4.2 NumPy

Used for numerical computations, especially while handling vector values from TF-IDF.

4.3 Matplotlib

Used for data visualisation such as bar charts representing ingredient frequency.

4.4 re (Regular Expressions)

Used for cleaning text by removing special characters and normalising ingredient names.

4.5 Scikit-learn

Key machine learning library used for: - TF-IDF Vectorization - Cosine Similarity - K-Means Clustering

4.6 WordCloud

Used to generate visual representation of ingredient importance using size-based display.

5. Data Preprocessing

The ingredients were converted to lowercase and cleaned using regular expressions to remove unwanted characters. They were then split into lists for better analysis. This step ensures uniformity and avoids duplication due to inconsistent naming (e.g., Salt vs salt).

Code snippet example:

```
df['Ingredients'] = df['Ingredients'].str.lower()
df['Ingredients'] = df['Ingredients'].str.replace('[^a-zA-Z, ]', '',
regex=True)
```

This preprocessing improved the accuracy of later analysis steps.

6. Ingredient Frequency Analysis

Using Python's Counter function, the most common ingredients across all dishes were identified. This helped understand which ingredients form the base of Uttarakhand cuisine.

A bar chart was generated showing the Top 10 Most Used Ingredients such as mustard oil, salt, garlic, cumin seeds, and turmeric.

This analysis transforms raw text into quantitative information.

7. TF-IDF and Ingredient Weighting

TF-IDF (Term Frequency-Inverse Document Frequency) was used to assign importance scores to ingredients.

This method highlights ingredients that: - Appear frequently in a specific dish - But rarely across all dishes

This helps distinguish unique ingredients from common ones. Ingredients like hemp seeds or fenugreek gain higher importance compared to salt or water.

Code example:

```
tfidf = TfidfVectorizer()
tfidf_matrix = tfidf.fit_transform(df['Ingredients'])
```

This process converts text into numerical vectors, enabling machine learning analysis.

8. Similar Dish Recommendation System

A recommendation engine was created using cosine similarity. It compares vectorised ingredient data to find dishes with similar compositions.

Function:

```
def recommend_similar_dishes(dish_name):
    similarity = cosine_similarity(tfidf_matrix[idx], tfidf_matrix).flatten()
```

This allows the system to recommend dishes that are similar in taste and ingredient profile, making it similar to real-world platforms like Swiggy or Zomato.

9. Clustering using K-Means

Unsupervised learning was applied using K-Means to group dishes into clusters based on ingredient similarity. Each dish was assigned a cluster number representing its category.

This helps automatic categorization without manual tagging, showcasing the power of machine learning.

10. Word Cloud Visualization

Ingredient Word Cloud was generated where the size of each ingredient is proportional to its frequency and TF-IDF importance.

This makes complex data visually understandable and aesthetically engaging.

11. Real-World Applications

This project can be applied to: - Food recommendation systems - Restaurant menu categorization - Cultural cuisine analysis - Diet-based meal recommendation apps - AI-based food discovery platforms

It demonstrates how text mining can enhance user experience by automating decision-making.

12. Advancements Over Basic Search

Earlier approach used keyword matching while the advanced approach uses: - Vectorization - Similarity scoring - Clustering

This changes the system from a static search tool to a smart recommendation engine.

13. Shortcomings of the Project

13.1 Dataset Limitations

- Small dataset size limits result accuracy
- Lack of ingredient quantity data
- Inconsistent ingredient naming
- No regional variation tagging
- Limited number of dishes

13.2 Technical Limitations

- Does not consider taste or texture
- No user feedback loop
- No real-time data source
- TF-IDF ignores semantic meaning

13.3 General Limitations

- Manual dataset creation can lead to bias
- No integration with external APIs

14. Scope for Improvement

- Expanding dataset size
- Adding nutritional information
- Introducing multi-language processing
- Adding ingredient synonyms
- Creating a web interface
- Applying deep learning models

15. Conclusion

This Text Mining Project successfully demonstrates how traditional recipe data can be transformed into structured intelligence using NLP and machine learning. It highlights the transition from simple data querying to intelligent recommendation and clustering systems.

Despite dataset limitations, the project effectively showcases how text mining can solve real-life problems and enhance decision-making in food technology platforms.

It stands as a practical demonstration of how data science can preserve culture while enhancing accessibility.

16. References

- Scikit-learn Documentation
- Python Pandas Official Guide
- Uttarakhand Traditional Recipe Sources
- NLP Theory Resources

Text Mining Project By Krishaang Anand