Dish Nutrition Simplifier: A Web-based Nutritional Analysis Tool

Akshat Raj Saxena, Aditya Sharma, Kanishk Kumar Meena

December 15, 2024

Contents

1 Title

Dish Nutrition Simplifier: A Web-based Nutritional Analysis Tool

2 Introduction

DishIT is a dish Nutrition Simplifier which is a comprehensive web application designed to analyze and optimize the nutritional profile of recipes. Integrates data-driven nutrient dense dataset sourced from USDA website having an abundance of nutritional profile for respective ingredients present and advanced algorithms for calculations of nutritional profile and suggesting top ingredients for a given protein, carbs and Sugar intake, the project aims to simplify complex nutritional data for everyday users while offering a robust platform for dietary planning and analysis.

3 Literature Review

Nutritional analysis and recipe optimization have gained significant importance in promoting health and wellness. Studies such as those from the USDA FoodData Central highlight the critical role of datasets in understanding food composition. Similarly, advanced data analysis techniques and web technologies have been instrumental in developing user-friendly tools for dietary management. This project builds upon these foundations, combining comprehensive datasets with innovative methodologies to deliver actionable insights.

4 Dataset

The dataset for this project was obtained from the **USDA FoodData Central**, a rich repository of nutritional information. The data includes details such as food descriptions, macronutrients (protein, carbohydrates, sugars), and other nutritional components per 100g. The downloaded CSV file was preprocessed and cleaned to ensure consistency and usability.

5 Methodology

The project workflow consists of the following steps:

5.1 Preprocessing and Cleaning

- Downloaded the dataset in CSV format from the USDA FoodData Central website.
- Parsed the CSV file using Python libraries such as `pandas` and `numpy`.
- Removed invalid rows (e.g., missing values or outliers).
- Extracted relevant columns: `Food Description`, `Protein (g)`, `Carbohydrates (g)`, and `Total Sugar (g)`.

5.2 Nutritional Profile Analysis

To determine ingredients with high nutrient content:

- Users input their daily nutrient intake (protein, carbohydrates, and sugar).
- The system filters ingredients from the dataset that meet or exceed these thresholds.
- Top 10 ingredients for each nutrient are identified and displayed.

5.3 Ingredient Substitution Logic

The application recommends alternative ingredients based on:

- Similar nutritional profiles to the user-selected ingredient.
- Suitability for specific dietary preferences or restrictions.
- Rankings based on nutrient density and culinary relevance.

5.4 Implementation

The backend was developed in Python with the Flask framework, while the frontend leverages React.js for dynamic user interaction. Data visualization and analysis were powered by `pandas`, `numpy`, and `react-bootstrap-typeahead`. The back-end implementation consists of a large dataset which is being used as a first hand for determination of nutrition result.

Frontend redirects users to a homepage, this implementation is done by App.js file which is referencing '/' page as homepage having implementation in Introduction.js. This hompeage consists of various functionality like knowing about Dataset, redirects to the main content, cantact us and a special functionality. All these file content functionality is implemented in components folder present in src directory.

6 Results

The application successfully:

- Identifies nutrient-dense ingredients across various categories and with their proper ingredient form.
- Provides detailed nutritional analysis of user-input recipes and if user want to include some healthy ingredients in their diets within a specific nutrient intake
- Suggests alternatives to improve nutritional value without compromising the dish's flavor profile.

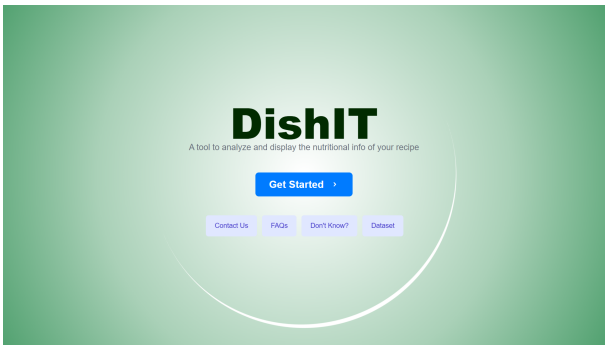

Figure 1: Starting Page

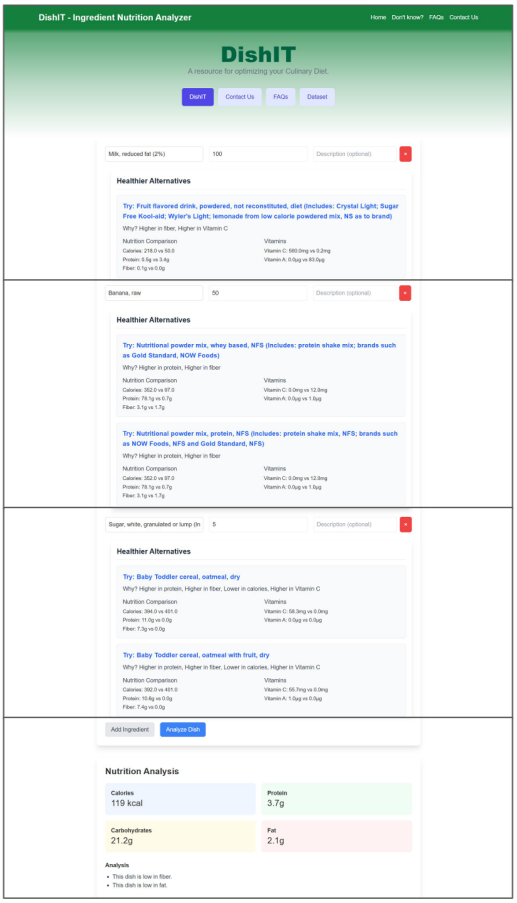

Figure 2: Main Page

7 Webserver

The application runs on a lightweight Flask web server:

- API endpoints handle ingredient submissions and return nutritional analyses.
- A React-based frontend communicates with the backend to deliver a seamless user experience.
- The `react-bootstrap-typeahead` component powers the dynamic dropdown for ingredient selection.

8 Future Work

The project has significant potential for future enhancements:

- Expand the dataset to include regional and less-common ingredients.
- Integrate machine learning models to predict recipes based on user dietary goals.
- Incorporate additional dietary filters (e.g., vegan, gluten-free, low-fat).
- Develop a mobile application for wider accessibility.

9 References

- USDA FoodData Central: `https://fdc.nal.usda.gov/`
- Python Documentation: `https://docs.python.org/3/`
- React Bootstrap Typeahead: `https://github.com/ericgio/react-bootstrap-typeahead`