Northeastern UNIVERSITY

Intelligent Healthy Recipe Generation App

Yujie Zhang

July 15, 2024

Northeastern UNIVERSITY

Introduction

- This project aims to generate personalized healthy recipes based on users' taste preferences and dietary restrictions.
- Improve users' dietary quality and health.
- Help users adhere to healthy eating habits through personalized recommendations.

Northeastern UNIVERSITY

Project Description

1. **Project Description**

- Features: Users input their dietary preferences and restrictions, and the app generates personalized recipes. Users can choose daily, weekly, or monthly meal plans based on the suggested recipes.
- Intelligence: Utilizes AI algorithms and big data analysis, combining food and nutritional information from multiple data sources to achieve intelligent recommendations.

Project Description

2. Advantage of Project

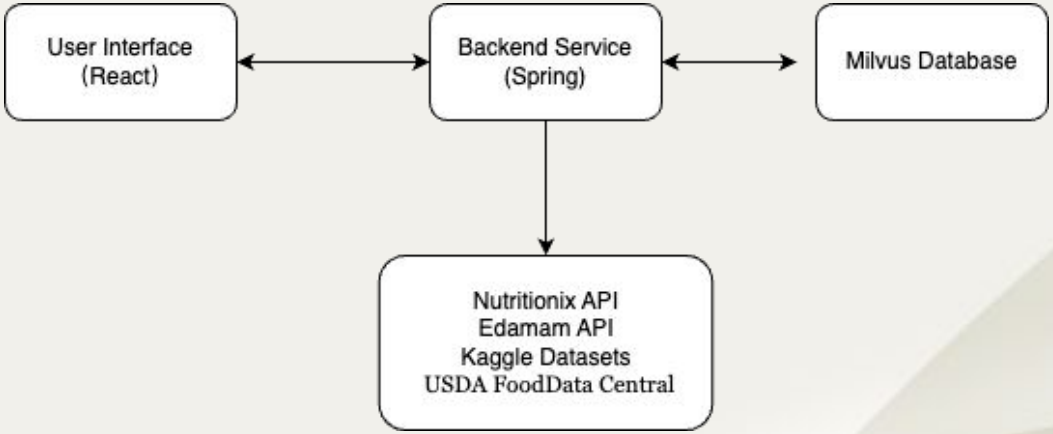
- Dietary Health Issues: Helps users select healthier meals to avoid malnutrition or overeating through personalized recommendations.
- Time Saving: Automatically generates recipes and shopping lists, saving users' time and effort.
- Adapting to Diverse Needs: Provides customized solutions based on users' dietary restrictions and preferences.

3. Scope of the Project

- Includes the development and integration of user interface, backend services, database, and recommendation engine.
- Plans to expand to mobile application platforms in the future.

Project Architecture

1. Architecture



- User Interface: Collects user preferences and displays recommended results.
- Backend Services: Handles user requests, performs data processing, and runs the recommendation algorithm.
- Milvus Database: Efficiently stores and retrieves recipe data.

2. Technologies and Tools Used

- Frontend: React.js
- Backend: Spring
- Database: Milvus
- Cloud Platform: AWS

Northeastern UNIVERSITY

Data Collection and Preprocessing

1. **Data Sources**
   - Nutritionix API: Provides detailed food and nutritional data.
   - Edamam API: Another widely used food and nutritional data source.
   - USDA FoodData Central: Offers comprehensive nutritional data from the USDA.
   - Kaggle Datasets: Utilize existing recipe and nutritional datasets from Kaggle.

2. **Data Collection Steps**
   - API Calls: Write Python scripts to fetch data from various APIs.
   - Data Storage: Store the retrieved data locally or in a cloud database for further processing.

3. **Data Preprocessing**
   - Data Cleaning: Remove duplicates, null values, and incomplete records.
   - Data Standardization: Convert data to a uniform format, such as standardizing units and field names.
   - Feature Extraction: Extract key features like ingredients, nutritional information, and cooking time.

RAG Pipeline

1. **User Preference Modeling**
   - Create user preference models based on their input regarding taste preferences and dietary restrictions.
   - Collect detailed user preference information through forms or surveys.
2. **Similarity Calculation**
   - Vectorize recipe data and user preferences (using TF-IDF or Word2Vec).
   - Use Milvus' vector similarity search feature to calculate the similarity between recipes and user preferences.
3. **Recommendation Algorithm**
   - Content-based Filtering: Recommend similar recipes based on ingredients and nutritional information.
   - Collaborative Filtering: Recommend recipes based on similar preferences of other users.
   - Hybrid Recommendation: Combine content-based and collaborative filtering methods to enhance recommendation accuracy.

Performance Metrics

1. **Key Metrics**
   - Response Time: Time taken to generate recommendations.
   - Accuracy: Relevance of recommended recipes and user satisfaction.
   - User Feedback: Users' ratings and reviews of recommended recipes.

2. **Calculation Methods**
   - Response Time: Measure the time from when the user submits a request to when they receive the recommendations.
   - Accuracy: Evaluate the relevance of recommendations through user feedback and A/B testing.
   - Collect user ratings for recommended recipes (1-5 stars).
   - Calculate the average rating.
   - User Satisfaction: Analyze user satisfaction through surveys and usage data.
   - Use Net Promoter Score (NPS) to measure user satisfaction.

Methods to Improve Metrics

1. **Improving Response Time**
   - Optimize Algorithms: Analyze and optimize the performance of recommendation algorithms to reduce computational complexity.
   - Caching Mechanism: Introduce a caching mechanism to store frequently requested results, reducing redundant computations.
   - Parallel Processing: Use parallel processing techniques to accelerate data processing and recommendation generation.
2. **Enhancing Accuracy**
   - Data Quality: Improve the quality and diversity of data, ensuring reliable and rich data sources.
   - Model Optimization: Optimize the recommendation model using advanced machine learning algorithms to increase accuracy.
   - User Feedback Loop: Establish a user feedback loop mechanism to continuously adjust and optimize the recommendation algorithm based on user feedback.
3. **Boosting User Satisfaction**
   - Personalized Recommendations: Further personalize recommendations by adjusting based on users' historical behaviors and feedback.
   - User Experience Optimization: Improve the user interface and interaction experience, making the app more user-friendly and intuitive.

Northeastern UNIVERSITY

Deployment Plan

1. **AWS Deployment Steps**
   - Set Up AWS Account: Register and configure an AWS account.
   - Deploy Backend Services: Use AWS Lambda or EC2 to deploy backend services.
   - Configure API Gateway to manage API requests.
   - Database Deployment: Deploy Milvus database on AWS (using AWS EC2 instances).
   - Frontend Deployment: Use AWS S3 and CloudFront to deploy static frontend files.

2. **Tools and Platforms**
   - AWS Lambda or EC2: For handling backend services.
   - AWS S3 and CloudFront: For hosting and accelerating frontend resources.
   - AWS RDS (Optional): For other data storage needs.

3. **User Testing and Feedback**
   - Internal Testing: Select a group of users for initial testing, gather feedback, and improve the app.
   - Public Testing: Expand the testing scope, collect more user data and feedback.

Northeastern UNIVERSITY

Future Work

1. **Expansion and Improvement**
   - Add more personalized options such as calorie calculation and meal planning.
   - Introduce machine learning models to enhance recommendation accuracy.
   - Develop a mobile app for convenient user access.
   - Example: Add voice input functionality, allowing users to input their dietary preferences via voice.
2. **Long-term Vision**
   - Expand the project to include more health management features such as exercise plans and health assessments.
   - Become a comprehensive health management tool for users.
3. **Next Steps**
   - Continue optimizing the recommendation algorithm to improve relevance and user satisfaction.
   - Expand data sources to include more recipes and nutritional data.
   - Collaborate with health experts to ensure the scientific accuracy and effectiveness of recommendations.

Conclusion

1. **Project Summary**
   - The Intelligent Healthy Recipe Generation App uses AI and big data analysis to generate personalized healthy recipes based on users' preferences and dietary restrictions.
2. **Key Takeaways**
   - Objective: Improve users' dietary quality and health.
   - Method: Utilize Nutritionix API, Edamam API, USDA FoodData Central, and Kaggle datasets for data, Milvus database for efficient storage and retrieval, and deploy on AWS cloud platform.
3. **Final Thoughts**
   - By continuously optimizing and expanding, the Intelligent Healthy Recipe Generation App can become a valuable tool for users to manage their health and achieve their dietary and lifestyle goals.

Northeastern UNIVERSITY

Q&A