Nutrient Recommendation System for Personalized Diet Planning

A report for Infosys Internship 5.0

by

Deepak Kumar
(Student of IIITDM KURNOOL)

Under the Guidance of

INFOSYS Mentor

November 2024

Infosys Internship 5.0

Title: Project Documentation: Nutrient Recommendation System for Personalized Diet Planning

Introduction:

The **Nutrient Recommendation System for Personalized Diet Planning** is a web-based application designed to help users make informed decisions about their diet and health by providing tailored meal plans. The system leverages data from user input (age, weight, height, dietary preferences, and health conditions) and nutritional information to generate weekly meal recommendations that align with the user's personal health goals.

Objectives:

- To provide users with personalized meal plans based on their nutritional needs and health conditions.
- To predict potential health risks or diseases based on a user's nutritional intake and medical background.
- To integrate machine learning models to offer health-focused recommendations, including weight management, diabetes, hypertension, and other health conditions.

Significance: This project addresses the growing need for personalized nutrition advice that can help individuals improve their health outcomes and manage chronic diseases through better dietary habits.

Project Scope:

Inclusions:

- **User Authentication**: A secure login and registration page.
- **Personalized Meal Plans**: Generation of weekly meal plans based on health conditions and preferences.
- **Disease Prediction**: Using machine learning models (Random Forest and Neural Networks) to predict potential diseases based on user input.
- **Dietary Preferences**: Filtering meal plans based on various dietary preferences like vegetarian, low-carb, etc.

Exclusions:

- The project does not include real-time health data tracking (e.g., integration with fitness trackers).
- Advanced user features like meal ratings or nutrition tracking are not included in this phase.

Limitations:

- The dataset is static, and meal suggestions are randomly selected, which may not be ideal for users with complex dietary restrictions.
- The machine learning models are trained on a specific dataset and may not cover all possible health conditions.

Requirements:

Functional Requirements:

1. **User Authentication**: Users should be able to register, log in, and access their personalized home page.
2. **Meal Plan Generation**: The app should generate a weekly meal plan based on user inputs (age, weight, height, health conditions).
3. **Disease Prediction**: Predict diseases based on the user's nutritional profile and health data.
4. **User Interface**: A clean and user-friendly interface for input forms and meal recommendations.

Non-Functional Requirements:

1. **Security**: All user data should be securely stored and protected using proper encryption techniques.
2. **Scalability**: The system should handle an increasing number of users and meal data.
3. **Performance**: The app should provide meal recommendations within a reasonable response time.

User Stories:

- As a user, I want to be able to log in and access my meal plan so that I can follow a diet that aligns with my health goals.
- As a user, I want to get personalized meal suggestions based on my health conditions, such as diabetes or hypertension.
- As a user, I want to see my BMI and get recommendations on whether I need to gain or lose weight based on my BMI.

Technical Stack:

Programming Languages:

- **Python**: Used for all development, including data processing, machine learning, and web interface logic.

Frameworks/Libraries:

- **Streamlit**: For creating the interactive web application and displaying results.
- **Pandas**: For handling and processing the nutrition data.
- **Pickle**: For loading pre-trained machine learning models.
- **Scikit-learn**: For building machine learning models (Random Forest) and preprocessing data.
- **TensorFlow/Keras**: For developing the neural network model for disease prediction.
- **IPython widgets**: For interactive widgets, such as text inputs, buttons, and forms.

Databases:

- **None**: The project uses static CSV files and in-memory data, though a database could be integrated in future versions.

Tools/Platforms:

- **Streamlit** : For development of the web application.

Architecture/Design:

Overview: The system is built with a modular design, consisting of multiple components:

1. **User Interface (Streamlit)**: Handles user inputs, displays recommendations, and interacts with the backend logic.
2. **Backend Logic**: Processes user inputs, handles authentication, filters the dataset, and generates meal plans.
3. **Machine Learning Models**: Used for disease prediction and health analysis (Random Forest and Neural Network).

Design Decisions:

- **Modularization**: The code is divided into functions that handle specific tasks (authentication, meal recommendations, model predictions, etc.), ensuring maintainability and scalability.

Development:

Technologies and Frameworks:

- **Streamlit**: Used for creating a dynamic user interface with interactive features like forms, checkboxes, and meal plan displays.
- **Pandas**: Used to manage and manipulate the nutrition data.
- **Pickle**: Used to load pre-trained models for disease prediction without retraining them in real-time.

Coding Standards:

- **Modular functions**: Each task is divided into functions for better maintainability.
- **Clear naming conventions**: Variable and function names are descriptive to improve code readability.
- **Error handling**: Proper error handling is implemented, especially in meal plan generation and user input processing.

Challenges:

- **Missing Data Handling**: Handling missing or incomplete meal suggestions by randomly selecting from the available data to avoid errors.
- **Model Optimization**: Fine-tuning the models for better accuracy in disease prediction. Models were trained on a fixed dataset, which might limit their generalizability.

Testing:

Testing Approach:

- **Unit Testing**: Each function was tested for correctness, including meal filtering, BMI calculation, and meal generation.
- **Integration Testing**: Ensured that the machine learning models work seamlessly with the data and the user interface.
- **System Testing**: The entire system was tested to ensure it behaves as expected, from logging in to receiving meal suggestions.

Testing Results:

- **Bugs**: Initial issues with random meal selection from missing values, which were resolved by filtering out `NaN` values.
- **Improvement**: Model performance in disease prediction could be improved by adding more training data.

Deployment:

Deployment Process:

Brief Overview of Deployment:

For now, the **Nutrient Recommendation System for Personalized Diet Planning** is deployed locally on my machine. Here's a summary of the deployment process:

1. **Install Dependencies:**
   - Install the necessary Python libraries like **Streamlit**, **Pandas**, **Scikit-learn**, and **TensorFlow** (if using machine learning models).
   - Use of virtual environment to manage dependencies effectively.
2. **Organize Project Files:**
   - Structure the project with relevant folders for data (`/data/`), models (`/models/`), and the main app file (`app.py`).
   - Ensure the nutrition data and machine learning models are accessible to the app.
3. **Running the Application Locally:**
   - Use the command `streamlit run app.py` to start the Streamlit server.
   - This launches the app on `http://localhost:8501`, which you can access through your browser.
4. **User Interaction:**
   - Users can log in, enter their details (age, weight, height, etc.), and receive personalized meal plans based on their health conditions and dietary preferences.
5. **Stopping the Application:**
   - To stop the app, use `Ctrl+C` in the terminal.

User Guide:

Instructions:

1. **Login**: Users must log in or register to access the personalized features of the app.
2. **Input**: After logging in, users can input their health data and preferences.
3. **Meal Plan**: Upon submission, the app will generate a weekly meal plan tailored to their needs.

Troubleshooting:

- **Login Issues**: Ensure the correct username and password are entered.
- **No Meal Plan Generated**: Check if the health conditions and dietary preferences match available data.

Conclusion:

The Nutrient Recommendation System successfully provides personalized meal plans and disease predictions to users based on their nutritional profile and health conditions. By combining machine learning with interactive web technologies, it offers a valuable tool for individuals looking to manage their health through diet.

Lessons Learned:

- Handling missing data effectively is crucial for maintaining the reliability of the system.
- Continuous improvement in the models and data quality will lead to better, more personalized recommendations.

Future Improvements:

- Incorporating real-time health data tracking (e.g., from fitness apps).
- Adding a more advanced meal recommendation engine that factors in nutritional analysis, taste preferences, and seasonal foods.