# Diet Recommendation System

## Project Overview

The **Diet Recommendation System** is a personalized web application designed to recommend dietary options based on an individual's unique characteristics, preferences, and health goals. The application employs a content-based approach and utilizes machine learning models to predict calorie requirements. The recommendations are further tailored to suit specific dietary preferences and medical conditions.

## Motivation

People from all around the world are getting more concerned about their health and way of life in today's modern environment. However, avoiding junk food and exercising alone are insufficient; we also need to eat a balanced diet. We can live a healthy life with a balanced diet based on our height, weight, and age. Your diet can help you achieve and maintain a healthy weight, lower your chance of developing chronic diseases (including cancer and heart disease), and improve your general health when combined with physical activity.

## What is a food recommendation engine?

A food recommendation engine using a content-based approach is an important tool for promoting healthy eating habits. This type of engine uses information about the nutritional content and ingredients of foods to make personalized recommendations to users. One of the key advantages of a content-based approach is that it takes into account an individual's dietary restrictions and preferences, such as allergies or food preferences. By providing users with tailored recommendations, a content-based food recommendation engine can help them make better choices about what to eat and improve their overall health. Additionally, by recommending a variety of healthy foods, it can also help users to discover new and nutritious options, expand their dietary horizons and overcome food boredom. All these can lead to a better and well-rounded diet, which can have a positive impact on long-term health outcomes.

### Objectives:

1. Predict the daily calorie intake required for users based on their personal data.
2. Provide meal recommendations categorized into **Breakfast**, **Lunch**, and **Dinner**.
3. Support dietary preferences and medical conditions to ensure suitability for the user.

4. Visualize the calorie distribution for better understanding.

### Tools and Technologies:

- **Programming Language:** Python
- **Libraries:**
    - Data Processing: Pandas, NumPy
    - Machine Learning: Scikit-learn
    - Visualization: Matplotlib
    - Web Application: Streamlit
- **Machine Learning Model:** Random Forest Algorithm

---

# Dataset Details

### Synthetic Dataset Creation:

A dataset with the following features was generated synthetically for this project:

| Feature | Description |
|---|---|
| Age | User's age (18-65 years) |
| Gender | Male or Female |
| Weight | Weight of the user (in kg, 50-120 kg) |
| Height | Height of the user (in cm, 150-200 cm) |
| Activity Level | Low, Moderate, or High |
| Dietary Preference | Vegan, Vegetarian, or Non-Vegetarian |
| Health Goal | Weight Loss, Maintenance, or Gain |
| Medical Condition | None, Diabetes, Hypertension, or Gluten Intolerance |
| Calories | Daily calorie requirement (1500-3000 kcal) |

---

# System Architecture

## 1. Machine Learning Model

**Data Preprocessing:**

- **One-Hot Encoding:** Applied to categorical features (e.g., Gender, Activity Level, Dietary Preference).
- **Target Variable:** Calorie requirements were set as the target for prediction.

**Model Training:**

- **Model:** Random Forest Regressor
- **Training:**
    - Split data into training (80%) and testing (20%) sets.
    - Trained the model on the preprocessed dataset.

**Model Saving:**

- Saved the trained model and feature columns using Pickle for reuse in the application.

## 2. Recommendation Logic

- Based on predicted calorie requirements, a food database was filtered by dietary preference and medical conditions to recommend suitable meals.

## 3. Web Application (Streamlit)

**Input:**

- User inputs including:
    - Age, Gender, Weight, Height
    - Activity Level, Dietary Preference
    - Health Goal, Medical Condition

**Output:**

- Predicted Calorie Intake
- Recommended Meals categorized as **Breakfast**, **Lunch**, and **Dinner**
- Calorie Distribution Visualization

---

# Features

## 1. Calorie Prediction

- Predicts daily calorie requirements based on user inputs using the trained Random Forest Regressor model.

### 2. Meal Recommendations

- Categorizes meals into:
    - **Breakfast**: 30% of total calories
    - **Lunch**: 40% of total calories
    - **Dinner**: 30% of total calories
- Tailors recommendations to dietary preferences and medical conditions.

### 3. Calorie Distribution Visualization

- Displays a bar chart showing calorie allocation across Breakfast, Lunch, and Dinner for better understanding of the diet plan.

---

# Implementation

### Step 1: User Input

- The application collects user inputs via sliders and dropdowns.

### Step 2: Data Processing

- Encodes the inputs to match the model's feature structure.

### Step 3: Calorie Prediction

- Predicts the daily calorie intake using the trained Random Forest Regressor.

### Step 4: Food Recommendation

- Filters food options based on:
    - User's dietary preference
    - Medical conditions

### Step 5: Visualization

- Displays:
    - Recommended meals categorized by type.
    - A bar chart of calorie distribution.

---

# Code Snippet Highlights

### 1. Calorie Prediction

```
# Predict calorie needs
predicted_calories = model.predict(input_data)[0]
st.subheader(f"Recommended Calorie Intake: {int(predicted_calories)} kcal")
```

### 2. Food Recommendation

```
def recommend_foods(calories, preference, condition):
    food_database = {
        "vegan": ["Quinoa Salad", "Avocado Toast", "Lentil Soup"],
        "vegetarian": ["Paneer Tikka", "Vegetable Stir Fry", "Cheese Sandwich"],
        "non-vegetarian": ["Grilled Chicken", "Salmon Salad", "Beef Steak"],
    }
    condition_filter = {
        "diabetes": lambda x: "low sugar" in x.lower(),
        "hypertension": lambda x: "low salt" in x.lower(),
        "gluten intolerance": lambda x: "gluten-free" in x.lower(),
        "none": lambda x: True,
    }
    foods = food_database[preference]
    return [food for food in foods if condition_filter[condition](food)]
```

### 3. Calorie Distribution Visualization

```
calorie_distribution = {
    "Breakfast": predicted_calories * 0.3,
    "Lunch": predicted_calories * 0.4,
    "Dinner": predicted_calories * 0.3,
}
fig, ax = plt.subplots()
ax.bar(calorie_distribution.keys(), calorie_distribution.values(), color=["blue", "green", "orange"])
ax.set_title("Calorie Distribution")
ax.set_xlabel("Meal Type")
ax.set_ylabel("Calories")
st.pyplot(fig)
```

---

# Future Enhancements

1. **Dynamic Food Database:** Integrate a larger, dynamic database of recipes with detailed nutritional values.
2. **Improved Categorization:** Use natural language processing (NLP) to classify meals based on descriptions.
3. **Mobile App Development:** Extend the application to mobile platforms for wider accessibility.
4. **Integration with Wearables:** Incorporate data from fitness trackers to improve calorie predictions.

---

# Conclusion

The **Diet Recommendation System** is an effective and user-friendly tool for personalized diet planning. By combining machine learning and Streamlit's interactive capabilities, the application provides actionable dietary recommendations and insights to users, helping them achieve their health goals efficiently.