
# CSE:587 Data Intensive Computing Project

---

## Analyze users' calorie intake to assess their diet and suggest improvements

---


### Team Members

Karthik Sharma Madugula - 50611293

Santosh Kota - 50593968

Riya Agarwal - 50609491

Harshita Sherla - 50593920

# 1 Introduction

This project will give personalized dietary insights by analyzing users' calorie intake. It will give recommendations to help improve eating habits and promote a better healthy lifestyle. This is crucial because good dietary habits are essential to preventing diseases like diabetes, obesity, and so on. Personalized recommendations based on food intake will help people to make healthier choices.

# 2 Methods

## 2.1 Data

We have used 2 datasets. The first dataset (user_nutritional_data.csv) contains information on the daily nutritional intake of approximately 2,000 users. It consists of survey data collected from residents and peers.

**Dimensions**: 2,182 rows and 11 columns

**Features**:

• Gender: Male (0) or Female (1) - Indicates the subject's gender.

• Age: Ranges from 15 to 75 years- Represents the subject's age, rounded down to the nearest integer.

• Daily Meal Frequency: Ranges from 2 to 4 - Indicates the average number of daily meals consumed by the subject.

• Physical Exercise: Ranges from 0 to 4 - Represent the level of daily exercise, with 0 indicating no exercise and 4 indicating extremely heavy exercise.

• Height: Ranges from 122 cm to 188 cm - The subject's height was recorded during the survey.

• Weight: Ranges from 35 kg to 150 kg - The subject's weight was recorded during the

survey.

• BMR: Ranges from 862 to 2,410 kcal - The Basal Metabolic Rate is calculated based on age, height, and weight.

• Carbs: Ranges from 129 g to 461 g - The total daily carbohydrate intake of the subject.

• Proteins: Ranges from 51 g to 184 g - The total daily protein intake of the subject.

• Fats: Ranges from 34 g to 123 g - The total daily fat intake of the subject.

The second dataset (nutrients.csv) provides comprehensive details on the nutritional composition of 14,164 food items, including macronutrients, micronutrients, and additional attributes.

**Dimensions**: 8,901 rows and 53 columns

**Features**:

Key columns include: NDB_No, Shrt_Desc, Calories, Protein_(g), Lipid_Tot_(g), Carbohydrt_(g), Fiber_TD_(g), Sugar_Tot_(g), Calcium_(mg), Iron_(mg), Vitamin C (mg), among others.

Other columns provide information on macronutrients, vitamins, minerals, and serving sizes.

## 2.2 Algorithms

For the hypothesis 1 (Riya Agarwal - 50609491): Physical exercise frequency is linked to higher protein intake among age groups, with BMR also influencing protein consumption.

The algorithm relies on:

1. Support Vector Machine (SVM):

The code uses a Support Vector Classifier (SVC) with a linear kernel to classify users' physical activity levels (Low or High) based on macronutrient intake (Carbs, Proteins, Fats). SVM is a supervised learning model that finds the hyperplane separating data into distinct classes by maximizing the margin between them.

2. Dimensionality Reduction with Truncated SVD:

To reduce the feature space from 3 dimensions (Carbs, Proteins, Fats) to 2 dimensions,

Singular Value Decomposition (SVD) is applied. SVD helps project the data into a lower-dimensional space, preserving as much variability as possible, making the data suitable for visualization.

3. Standardization:
Data is scaled using StandardScaler to normalize the features (Carbs, Proteins, Fats) to have zero mean and unit variance. This ensures that all features contribute equally to the analysis.

In the code:

1. Data Preprocessing: The dataset includes columns for macronutrients (Carbs, Proteins, Fats) and physical activity levels (Physical exercise). Physical activity levels are mapped to binary classes:

$$\text{Low} = \{0, 1\}, \quad \text{High} = \{2, 3, 4\}.$$

   Features are scaled to have zero mean and unit variance.

2. Dimensionality Reduction: SVD is applied to reduce the dimensionality of the standardized macronutrient data to two components:

$$\mathbf{X}_{\text{SVD}} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^T,$$

   where $\mathbf{X}_{\text{SVD}}$ represents the reduced 2D feature space.

3. Model Training: An SVM classifier with a linear kernel is trained on the reduced feature space to classify physical activity levels. The model learns to separate Low and High activity levels by maximizing the margin between data points.

4. Visualization: A scatter plot of the reduced data illustrates clusters for Low and High physical activity levels based on macronutrient intake.

5. User Input and Prediction: Users input their daily macronutrient intake (Carbohydrates, Proteins, Fats). The input is scaled, reduced using SVD, and classified by the SVM model to predict whether the user's diet aligns with Low or High physical activity levels.

6. Macronutrient Analysis: The app calculates the total calorie intake:

$$\text{Calories} = (\text{Carbs} \times 4) + (\text{Proteins} \times 4) + (\text{Fats} \times 9).$$

The percentage distribution of macronutrients is computed:

$$\text{Carbohydrates } \% = \frac{\text{Carbs} \times 4}{\text{Total Calories}} \times 100,$$

$$\text{Proteins } \% = \frac{\text{Proteins} \times 4}{\text{Total Calories}} \times 100,$$

$$\text{Fats } \% = \frac{\text{Fats} \times 9}{\text{Total Calories}} \times 100.$$

For the hypothesis 2 (Harshita Sherla - 50593920): The interaction between macronutrients and specific micronutrients significantly influences the overall health benefits of food items, as measured by a health score that considers caloric content, nutritional density, and micronutrient adequacy.

The algorithm relies on:

1. Random Forest Regressor: A supervised machine learning model is employed to predict a health score based on the interaction of macronutrients and micronutrients. Random Forest is chosen for its robustness to overfitting and its ability to handle nonlinear relationships.

2. Feature Engineering: Interaction terms are created between specific nutrients to capture complex relationships:

Protein x Calcium: Measures the combined effect of protein and calcium.

Fiber x Iron: Evaluates the combined impact of fiber and iron on health.

In the code:

1. Nutrient Data Preparation: A subset of numerical nutrient columns is selected from the dataset:

Macronutrients: Protein (g), Fiber (g), Saturated Fats (g).

Micronutrients: Calcium (mg), Iron (mg), Vitamin C (mg), Cholesterol (mg).

Interaction features are engineered:

Protein x Calcium and Fiber x Iron are computed and added as new columns.

2. Health Score Calculation A custom function (calculate_health_score) is defined to compute a health score for each food item:

Health Score = $2 \times$ Protein + $1.5 \times$ Fiber + $0.1 \times$ Calcium + $0.1 \times$ Iron + $0.2 \times$ Vitamin C $- 0.05 \times$ Cholesterol $- 0.2 \times$ SaturatedFats

This score weighs positive contributors (e.g., protein, fiber, calcium) and penalizes negative factors (e.g., saturated fats, cholesterol).

3. Random Forest Model Training

Features (X): Nutritional values, including interaction terms (Protein x Calcium, Fiber x Iron).

Target (y): The computed health score for each food item.

Training Process: Data is split into training and test sets (80/20 split). The Random Forest Regressor is trained on the training set to predict health scores.

4. Model Predictions: The model predicts health scores for test samples. The predicted score for a user-selected food item is displayed as: Predicted Health Score: [value]

5. Recommendations:

Feedback Based on Predicted Health Score:

- Score $> 75$: Diet is healthy; user is encouraged to maintain the same food choices.
- Score 50–75: Moderate health; suggestions include adding fruits, vegetables, and other nutrient-dense foods.
- Score $< 50$: Unhealthy diet; recommendations focus on cutting saturated fats and increasing high-fiber, protein-rich foods.

Recommendations for Nutrient Deficiencies:

Nutrients like protein, fiber, calcium, and iron are analyzed for deficiencies. Foods rich in these nutrients are recommended based on ranking.

For the hypothesis 3 (Riya Agarwal - 50609491): Food Comparison Tool (Takes different food as input and give nutrients present in the food, so that user can choose accordingly on what to eat.)

The algorithm relies on:

The code employs:

1. Data Subsetting: Filters user-selected food items from the nutrient dataset. Extracts relevant nutrient columns for comparison.
2. Visualization: Generates a bar chart comparing the nutritional values of selected foods for user-specified nutrients.

In the code:
1. User Input Handling:
Food Selection: Users are presented with a dropdown list of all available food items in the dataset (nutrients.csv) to select multiple items for comparison.
This is implemented using the Streamlit multiselect widget.
Nutrient Selection: Users can also select specific nutrients (e.g., Protein (g), Calories, Fiber (g)) for comparison from the dataset columns.
2. Data Processing:
Filters the dataset: Extracts rows corresponding to user-selected foods. Retains only the selected nutrients along with the food names.
3. Visualization:
Table: A dataframe showing the nutrient values of the selected foods is displayed in tabular format for quick reference.
Bar Chart: A bar chart is generated using the filtered dataset: Nutrient values are plotted for each food item.
Foods are displayed on the x-axis, and nutrient amounts are plotted on the y-axis.
4. Error Handling:
If no foods or nutrients are selected: A warning message is displayed to prompt the user to make selections.
5. Visualizations and Insights: A dataframe and plots display the relationship between:
- Exercise levels
- Protein intake
- Adjusted BMR
Recommendations for high-protein foods are made if protein intake is insufficient.

For the hypothesis 4 (Santosh Kota - 50593968): Dietary patterns can be classified into distinct types (e.g., Mediterranean, vegetarian, high-protein) based on the nutrient profiles of foods consumed, where nutrient diversity and ratios are key indicators of dietary adherence.

The algorithm relies on:

1. K-Nearest Neighbors (KNN):

A supervised machine learning algorithm used for classification. Based on nutrient ratios and nutrient diversity, the KNN model predicts the dietary pattern (High-Protein, Mediterranean, Vegetarian, or Standard).

It assigns the label of the nearest neighbors in the feature space using Euclidean distance.

2. Custom Rule-Based Labeling:

Dietary patterns are manually labeled using predefined nutrient thresholds:

- High-Protein: Foods with a high protein-to-carb/fat ratio.
- Mediterranean: High fiber-to-carb ratio and low sugar.
- Vegetarian: Low fat, high carbs.
- Standard: Balanced nutrients without specific focus.

In the code:

1. Nutrient Data Preprocessing:

Feature Selection: Nutrient columns such as Calories, Fat (g), Protein (g), Carbohydrate (g), Fiber (g), Sugars (g), Calcium (mg), Iron (mg), and Potassium (mg) are selected as features for classification.

Imputation: Missing nutrient values are filled using the mean imputation strategy with SimpleImputer.

2. Dietary Pattern Label Creation:

A custom function generates labels for each row of the dataset based on the following rules:

- High-Protein: Protein ratio ($\frac{\text{Protein (g)}}{\text{Carbohydrate (g)+Fat (g)}}$) is above 0.5.
- Mediterranean: Fiber-to-carb ratio ($\frac{\text{Fiber (g)}}{\text{Carbohydrate (g)}}$) is above 0.1, and sugar content is low.
- Vegetarian: Low fat and high carbohydrate content.

- Standard: Balanced but not meeting the above criteria.

3. Data Splitting and Scaling:

The dataset is split into training and testing sets (80/20 split). Features are scaled using StandardScaler to standardize data for KNN.

4. KNN Model Training:

The KNN classifier is trained on the scaled features with: n_neighbors=5 (considers the 5 nearest neighbors).

Distance metric: Euclidean distance.

5. Predictive Analysis for Selected Foods:

Prediction: For user-selected foods, their total nutrient values are calculated and scaled using the same scaling model. The KNN model predicts the dietary pattern based on the nutrient profile.

Comparison with Other Patterns: Nutrient profiles are grouped by dietary patterns in the dataset, and their means are calculated. A heatmap compares the user's total nutrient profile against the mean nutrient profiles of the different patterns.

6. Recommendations Based on Predicted Pattern:

Food recommendations are tailored to the predicted dietary pattern:

High-Protein: Focus on foods low in carbs and fats.

Mediterranean: Prioritize fiber and healthy fats while minimizing sugar.

Vegetarian: Suggest plant-based, low-fat, high-carb foods.

Standard: Recommend balanced nutrients across all categories.

For the hypothesis 5 (Karthik Sharma Madugula - 50611293): Categorizing foods into weight-gain and weight-loss categories based on calorie content and suggesting foods that align with individual dietary goals (e.g., achieving a target calorie intake).

The algorithm relies on:

1. HistGradientBoostingClassifier:

A supervised machine learning model from sklearn.ensemble is used for classification. It

classifies foods into weight-gain or weight-loss categories based on their calorie content and nutrient profiles.

2. Custom Rule-Based Filtering:
Foods are suggested based on whether the user needs to gain or lose weight: High-calorie foods for weight gain and Low-calorie foods for weight loss.

In the code:
1. Data Preprocessing:
Feature Selection: The following features are selected from the nutrient dataset for modeling: Calories, Fat (g), Protein (g), Carbohydrate (g), and Sugars (g). The categorical feature Food Group is one-hot encoded to include its influence on weight categorization.
Target Variable: Foods are categorized into weight-gain (1) and weight-loss (0) based on their calorie content:
A threshold of 250 calories is used:
Weight Gain: Foods with Calories $> 250$.
Weight Loss: Foods with Calories $\leq 250$.
Data Transformation: The dataset is transformed into numerical and categorical features using one-hot encoding for categorical variables (Food Group).
2. Model Training:
HistGradientBoostingClassifier: The model is trained on the preprocessed data to predict whether a food contributes to weight gain or loss. Gradient boosting iteratively improves predictions by reducing errors at each stage, making it robust for this classification task.
3. User Inputs:
Food Selection:
- Users select foods from the dataset for analysis.
- Total calories of the selected foods are calculated.
Target Calorie Goal: Users input their target calorie intake for the day. The system calculates the difference between the user's target calorie goal and the total calorie content of the selected foods:

- Remaining Calories $> 0$: User needs more calories to meet their goal.

- Remaining Calories $< 0$: User has exceeded their calorie goal.

4. Predictions and Recommendations:

Weight-Gain/Weight-Loss Prediction: The model predicts whether the selected foods contribute to weight gain or weight loss. If the predictions align with the user's goal (e.g., weight loss foods for a weight loss goal), the system acknowledges the selection. Otherwise, the system flags mismatched food choices.

Food Suggestions:

For weight gain: Foods are suggested that can help the user meet the remaining calorie requirement without exceeding it.

For weight loss: Foods with lower calories are recommended to help the user reduce their calorie intake.

5. Visualization:

Recommendations Table: A table lists the suggested foods with their calorie values. Foods are sorted by calorie content to help users make informed choices.

# 3 Results and Discussion

For the hypothesis 1: A clear relationship between macronutrient intake and physical activity levels, as visualized through distinct clusters in the scatter plot is seen. Users with higher physical activity levels tend to consume diets richer in proteins and balanced macronutrients, while those with lower activity levels exhibit less optimized distributions. The app effectively classifies physical activity levels based on dietary patterns and provides actionable recommendations, such as increasing protein intake and achieving a balanced macronutrient distribution. These insights enable users to align their dietary habits with their activity levels for improved health and performance.

For the hypothesis 2: Macronutrient-micronutrient interactions (e.g., Protein x Calcium) are key factors influencing the health score. Foods are ranked and suggested based on their ability to improve the health score and address deficiencies.

For the hypothesis 3: Users can quickly compare foods based on nutritional content, enabling informed dietary choices. The tool is especially helpful for identifying foods rich in specific nutrients (e.g., high-protein, low-fat options).

For the hypothesis 4: Nutrient ratios are critical in defining dietary patterns (e.g., high protein-to-carb ratio for High-Protein diets). Visualization (heatmaps) aids in understanding how the user's diet compares to other patterns. Personalized recommendations encourage adherence to a healthier dietary pattern.

For the hypothesis 5: Foods are categorized into weight-gain or weight-loss groups based on calorie content. The system dynamically adjusts recommendations based on the user's target calorie goal. The model ensures personalized and actionable suggestions, improving the user's dietary decisions.

# 4 Screenshots of the Application:

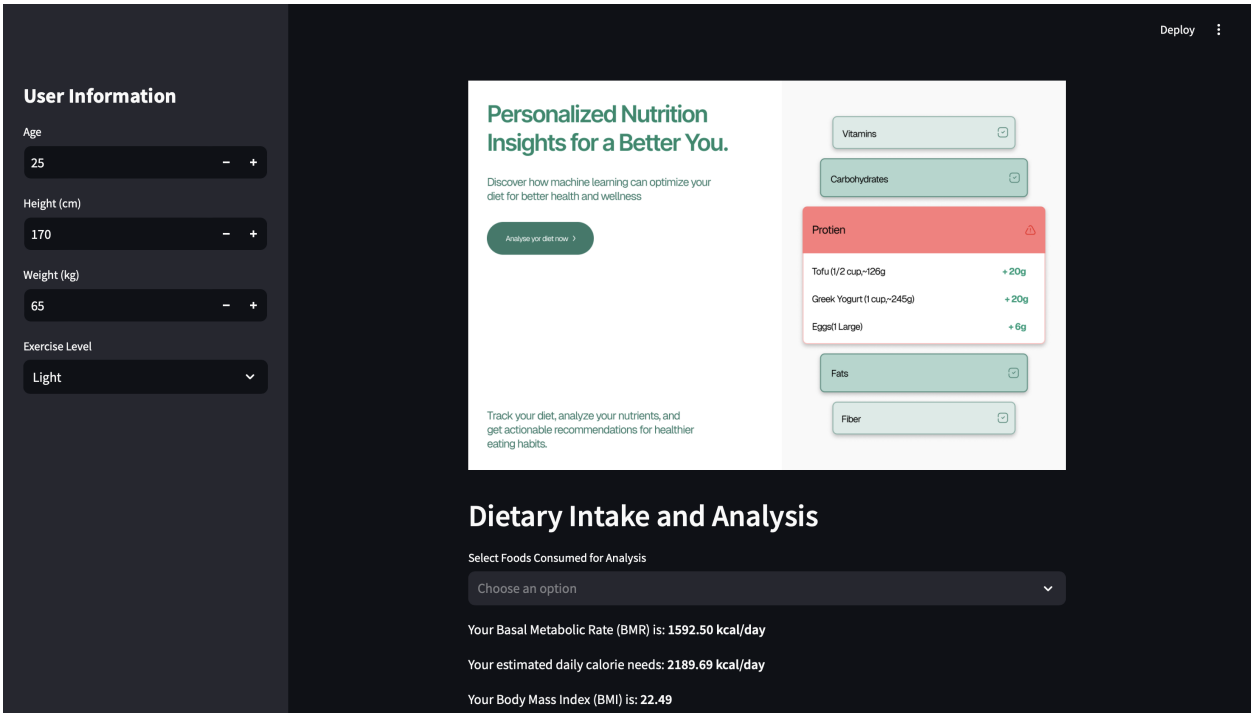


Figure 4.1: Screenshot 1

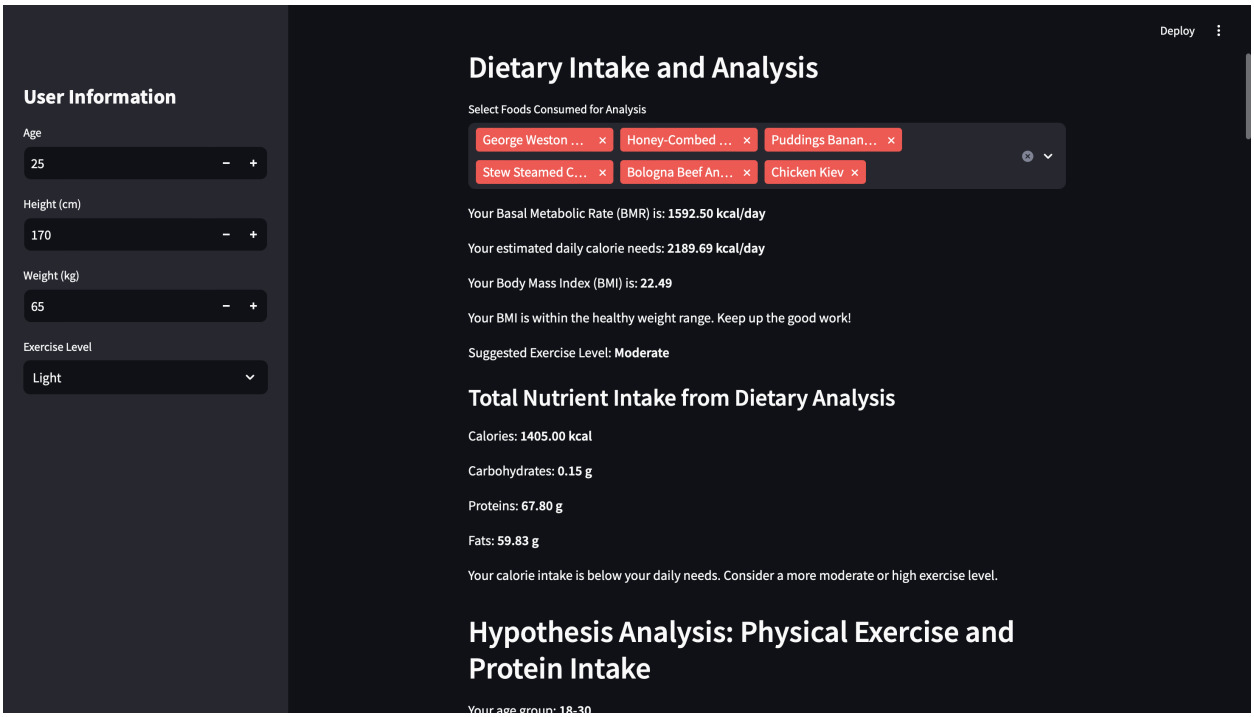


Figure 4.2: Screenshot 2

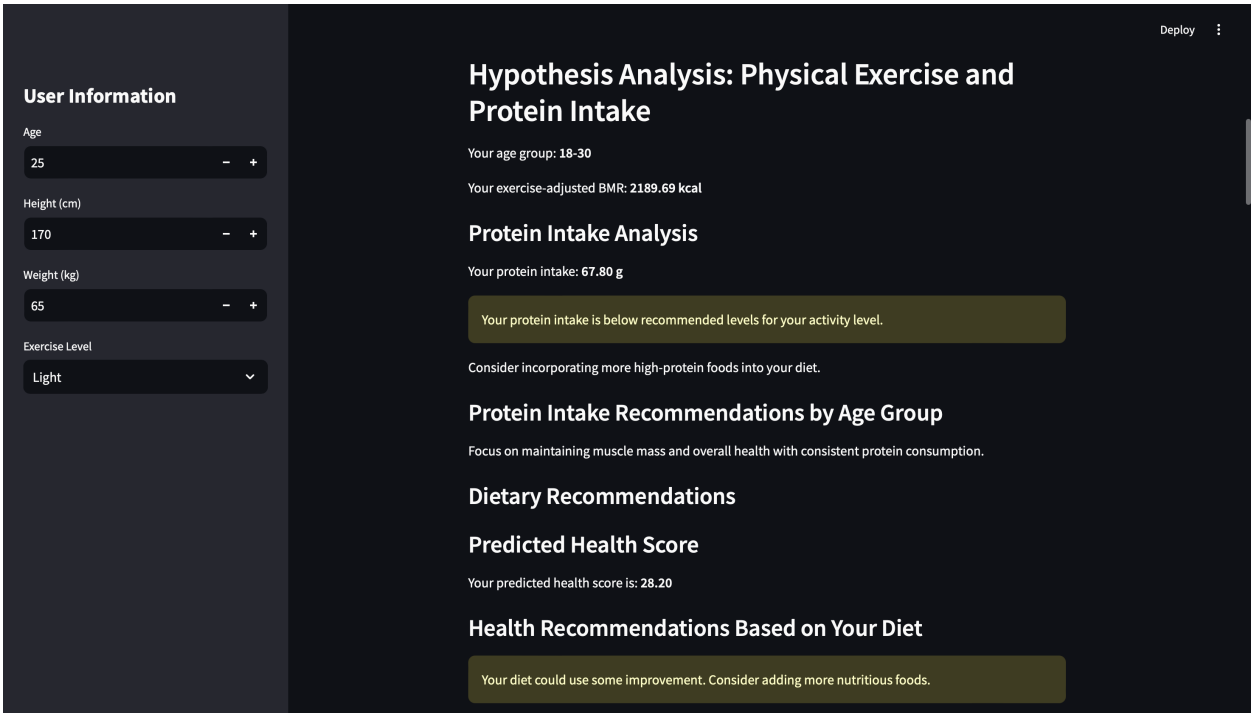


Figure 4.3: Screenshot 3

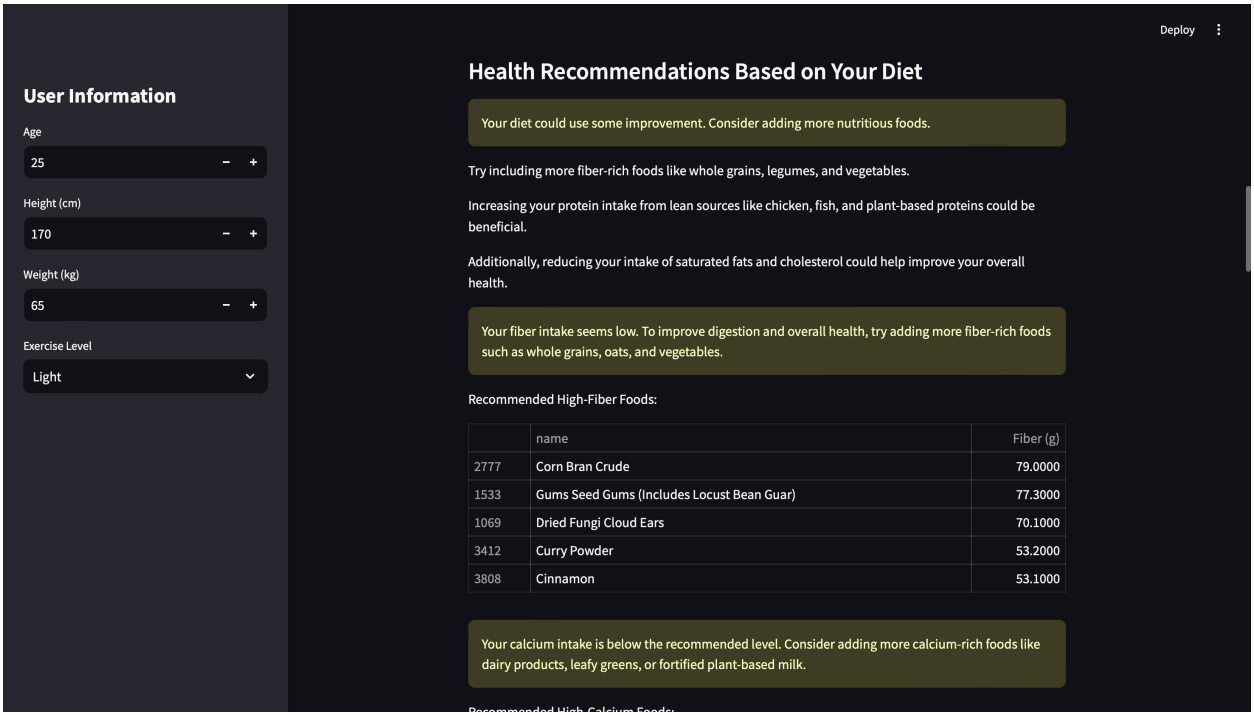


Figure 4.4: Screenshot 4

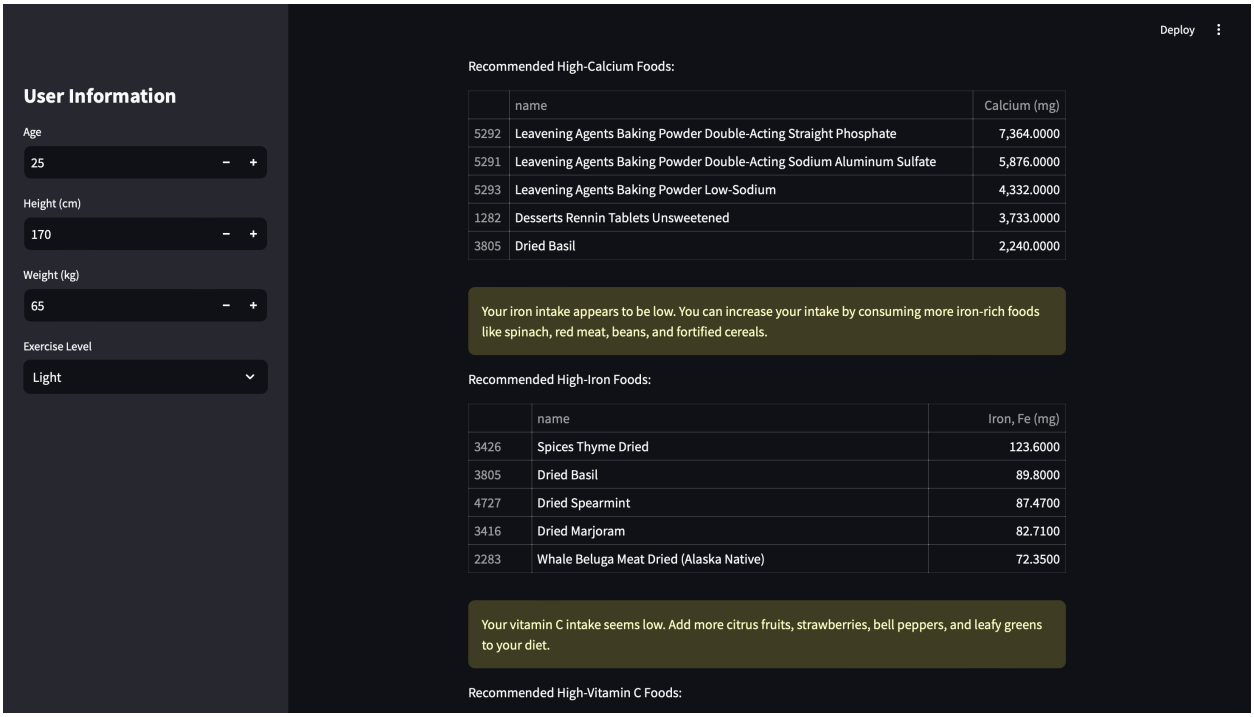


Figure 4.5: Screenshot 5

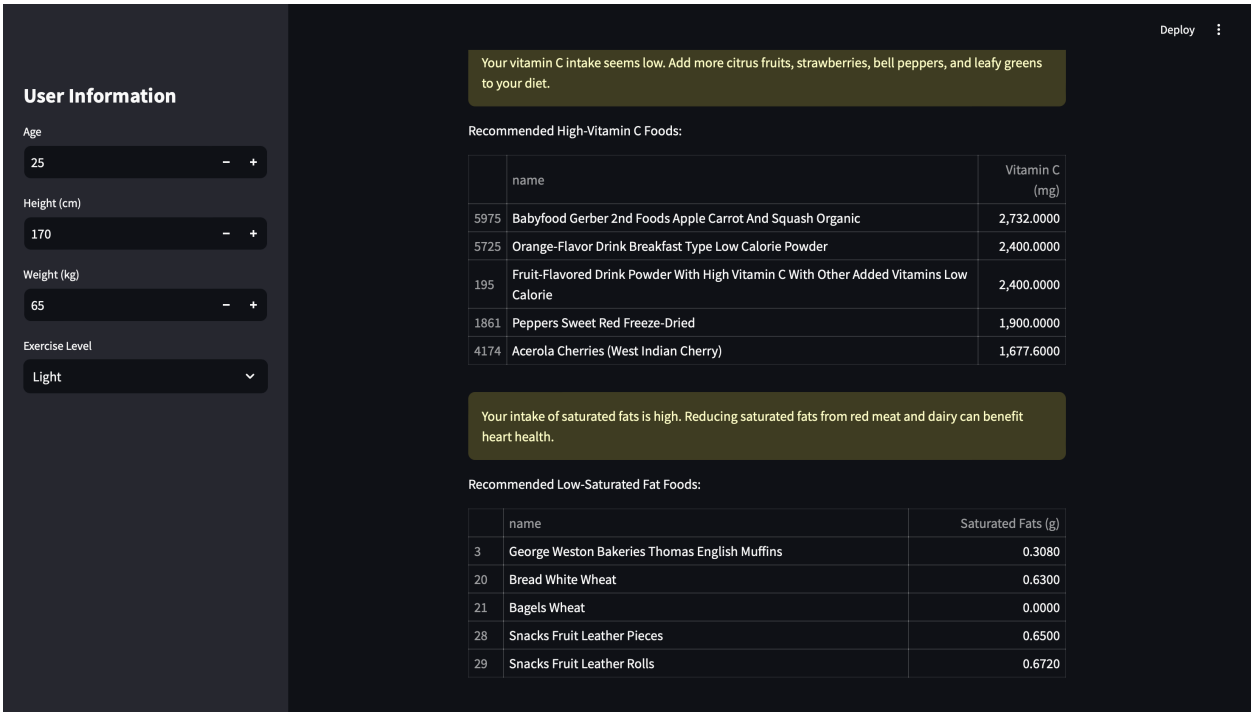


Figure 4.6: Screenshot 6

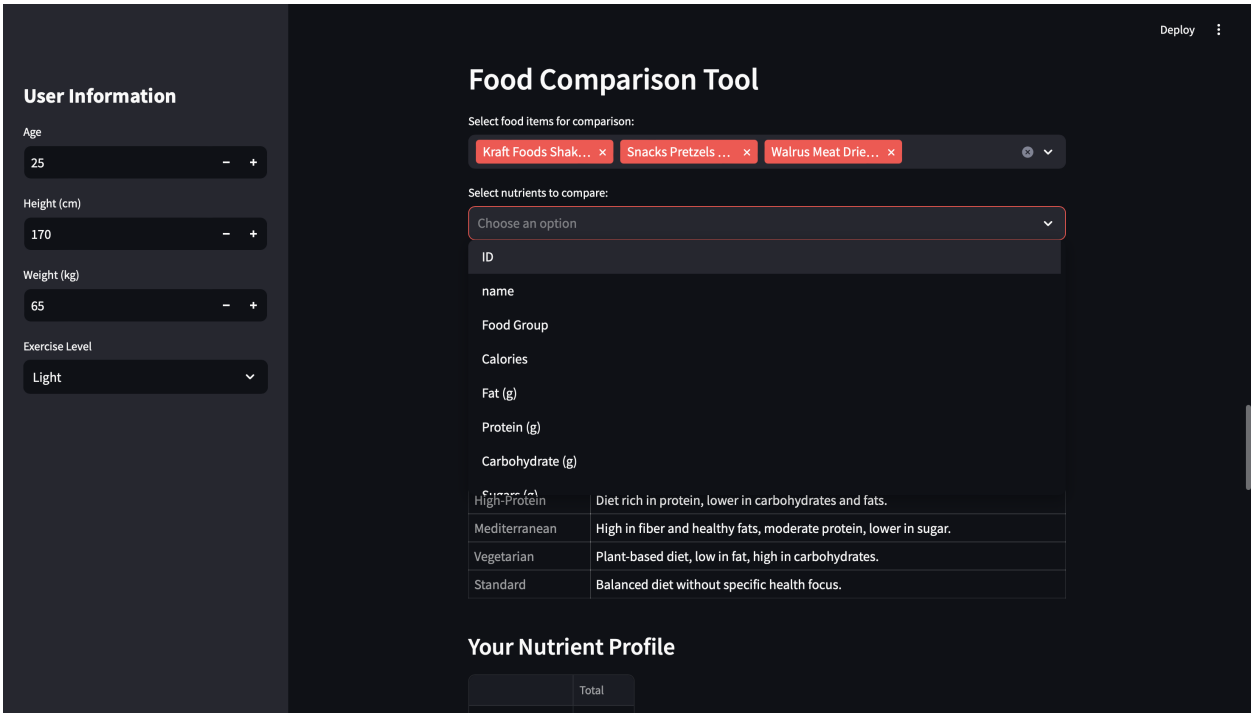


Figure 4.7: Screenshot 7

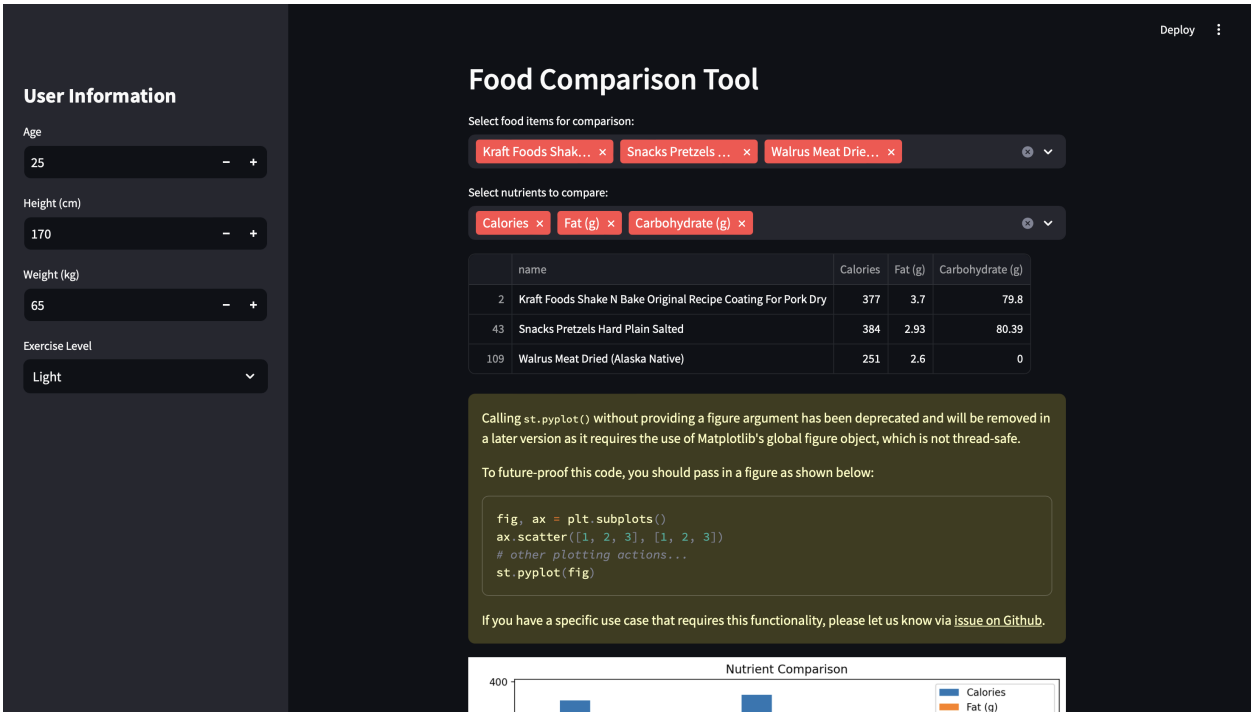


Figure 4.8: Screenshot 8

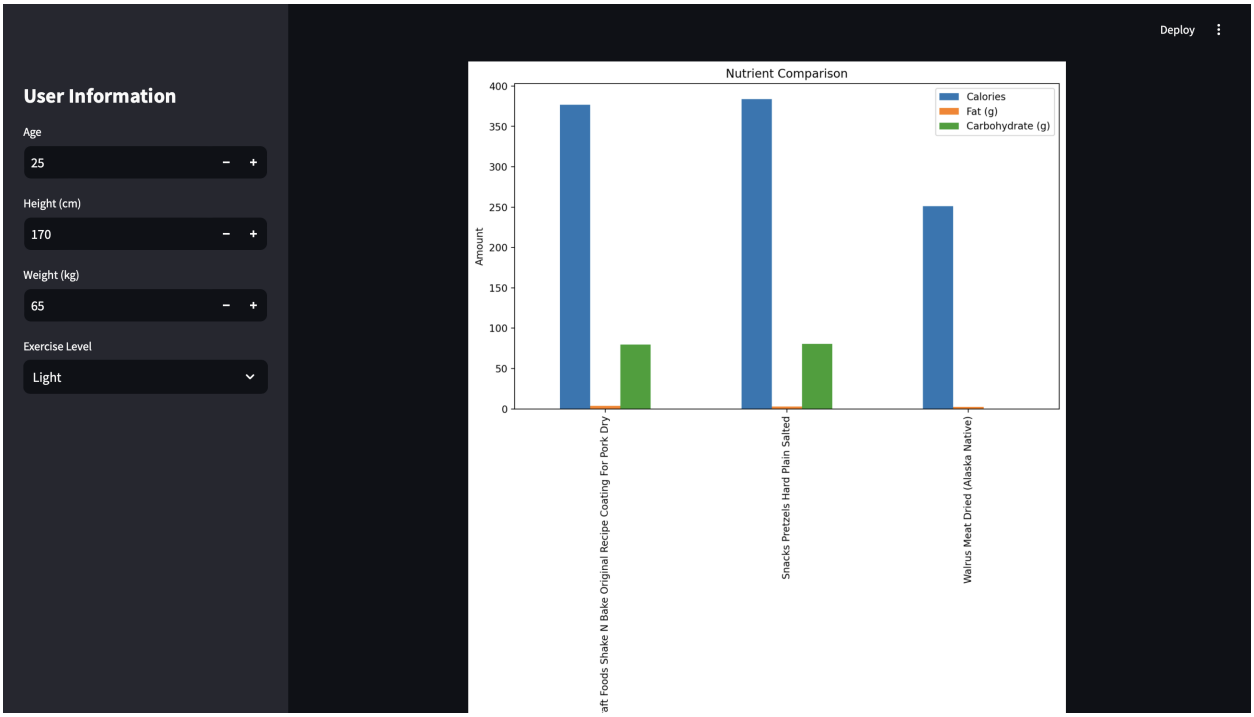


Figure 4.9: Screenshot 9

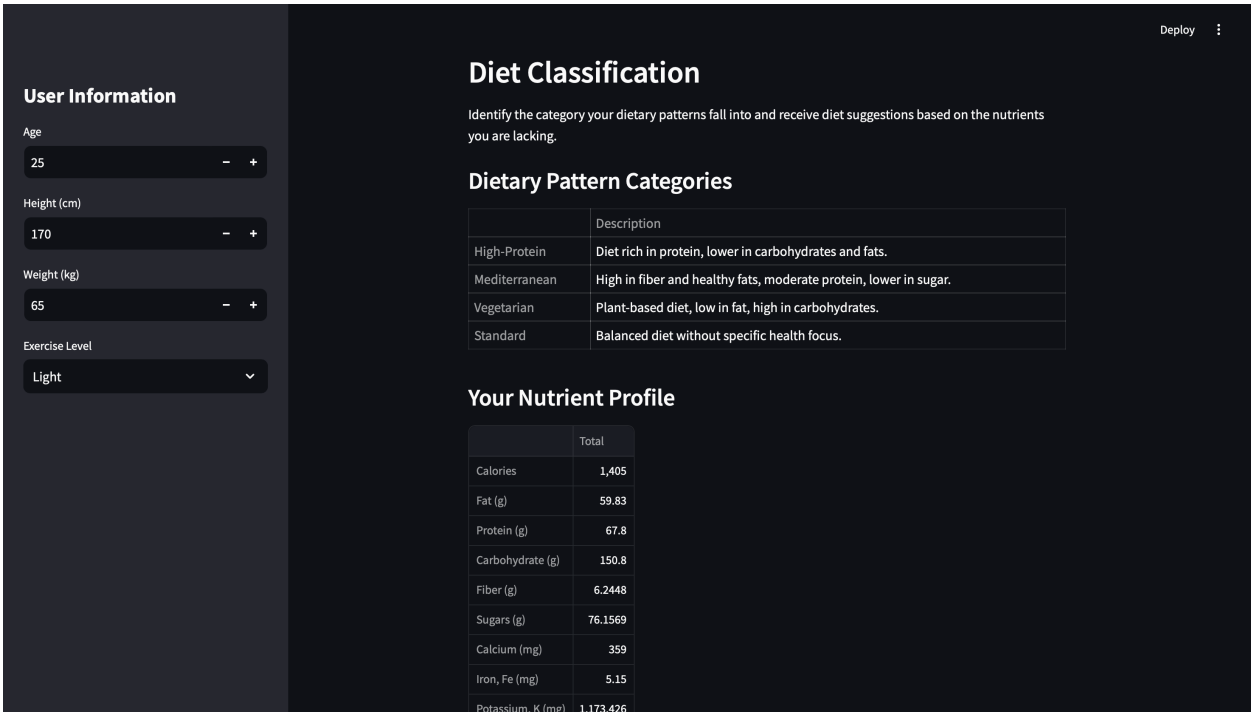


Figure 4.10: Screenshot 10


Figure 4.11: Screenshot 11

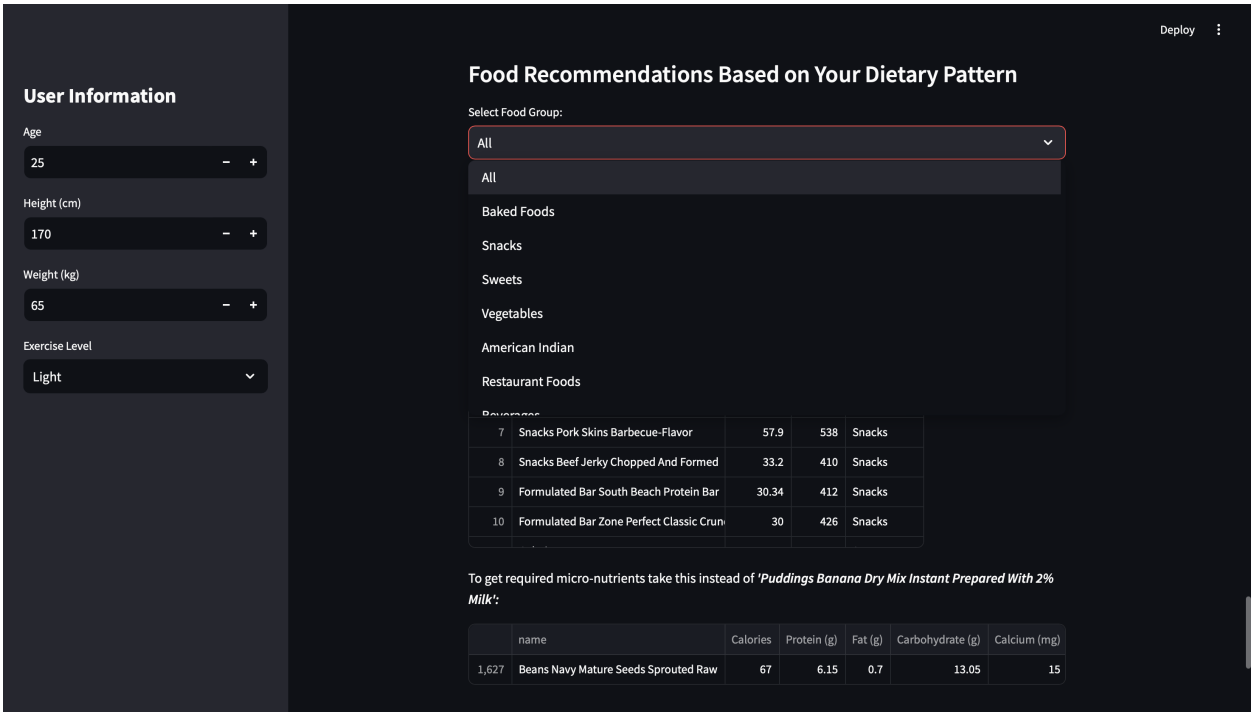


Figure 4.12: Screenshot 12

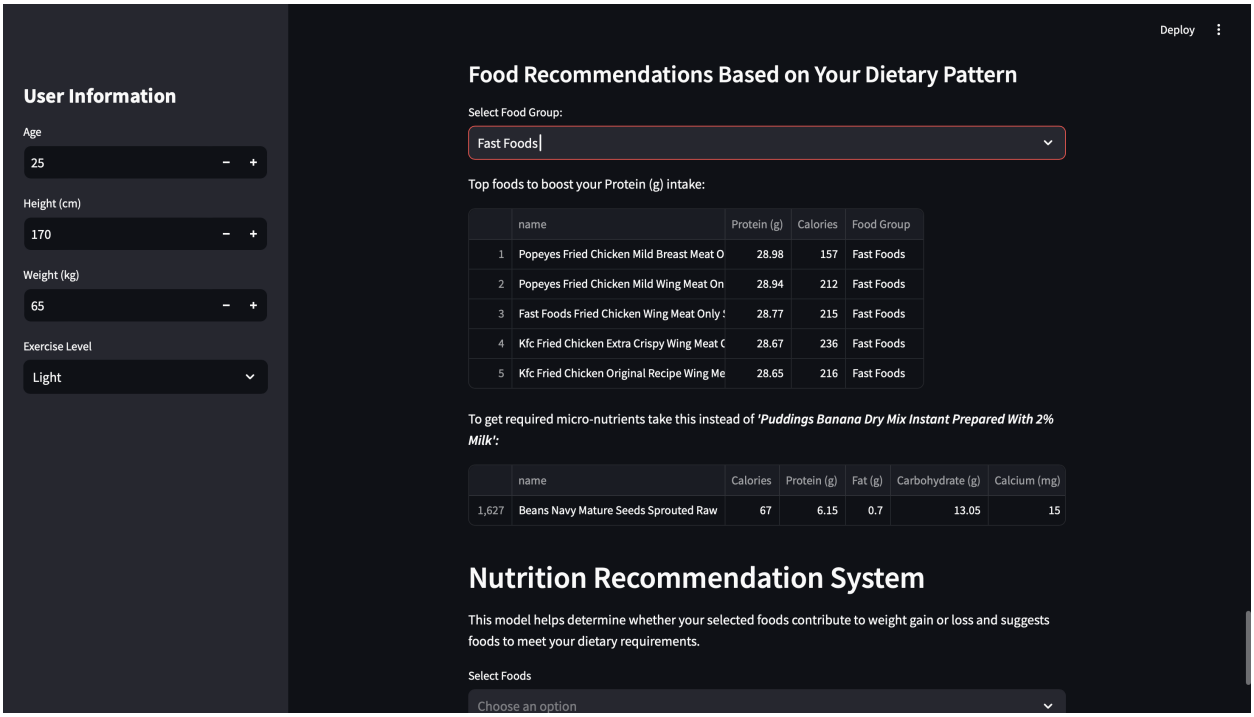


Figure 4.13: Screenshot 13

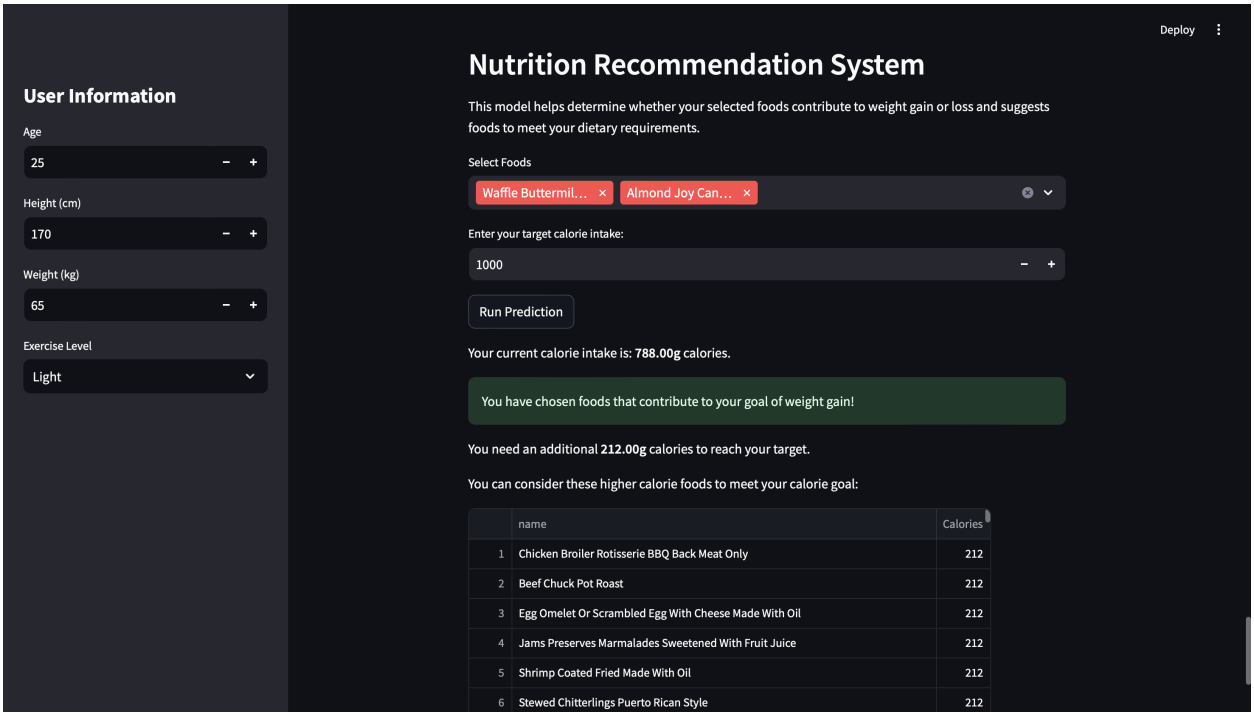


Figure 4.14: Screenshot 14

The app provides a comprehensive, user-centered platform for nutritional analysis, offering insights and recommendations based on user inputs and dietary patterns. The user interface is highly interactive, allowing users to input personal details such as age, weight, height, and activity levels, which are used to calculate key metrics like Basal Metabolic Rate (BMR) and Body Mass Index (BMI). The app categorizes foods and analyzes nutritional data in detail, classifying dietary patterns into categories like High-Protein, Mediterranean, Vegetarian, and Standard, as evident from the nutrient profile comparison heatmaps. This classification helps users align their dietary habits with healthier options. The protein intake analysis linked to exercise levels highlights the app's ability to identify deficiencies and suggest improvements specific to individual goals. The Food Comparison Tool enables users to compare nutrients across selected food items visually through bar charts, making it easier to make informed decisions. Additionally, the Nutrition Recommendation System evaluates whether users' food choices align with weight gain or weight loss goals based on calorie targets and provides tailored suggestions to help meet those goals. For instance, users are advised to incorporate high-calorie foods for weight gain or low-calorie foods for weight loss, as demonstrated in the screenshots. The app also identifies nutrient deficiencies like low vitamin C or high saturated fat intake, offering precise food recommendations to address these issues. Moreover, the ability to filter foods by group, such as Fast Foods or Snacks, makes the app practical for daily use while maintaining a focus on health. The visualizations, such as nutrient profile comparisons and the predicted health score, provide users with actionable insights, making the app not just a tool for tracking calories but also an intelligent system for improving dietary health holistically. Overall, the screenshots illustrate a powerful, data-driven approach that combines machine learning, nutrient analysis, and a user-friendly interface to encourage healthier eating habits effectively.

# Bibliography


[1] Abdus Samad. (2022). *User Daily Nutritional Intake*. Kaggle. https://www.kaggle.com/datasets/abdussamad123/user-daily-nutritional-intake

[2] U.S. Department of Agriculture and U.S. Department of Health and Human Services. (2020). *Dietary Guidelines for Americans, 2020-2025* (9th Edition). https://www.dietaryguidelines.gov/

[3] Institute of Medicine (US) Standing Committee on the Scientific Evaluation of Dietary Reference Intakes. (2005). *Dietary Reference Intakes for Energy, Carbohydrate, Fiber, Fat, Fatty Acids, Cholesterol, Protein, and Amino Acids*. The National Academies Press. https://doi.org/10.17226/10490

[4] Mifflin, M. D., St Jeor, S. T., Hill, L. A., Scott, B. J., Daugherty, S. A., & Koh, Y. O. (1990). A new predictive equation for resting energy expenditure in healthy individuals. *The American Journal of Clinical Nutrition,* 51(2), 241–247. https://doi.org/10.1093/ajcn/51.2.241

[5] National Institutes of Health (NIH): Office of Dietary Supplements. https://ods.od.nih.gov/

[6] World Health Organization (WHO). (2003). *Diet, Nutrition and the Prevention of Chronic Diseases.* WHO Technical Report Series, No. 916. https://www.who.int/publications/i/item/924120916X

[7] Streamlit Documentation. https://docs.streamlit.io/

[8] Python Documentation. https://docs.python.org/3/

[9] Pandas Documentation. `https://pandas.pydata.org/docs/`

[10] SQLite Documentation. `https://www.sqlite.org/docs.html`

[11] McKinney, W. (2012). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython.* O'Reilly Media.

[12] Westra, E. (2018). *Building Serverless Python Web Services with Zappa.* Packt Publishing.