Project Report

Title: *Diet and Recipe Recommendation System Using Nutrition-Based Content Filtering*

---

1. Introduction

Healthy eating requires choosing meals that align with specific nutrition goals — for example, high-protein diets for muscle gain or low-carbohydrate diets for diabetes management. However, manually finding such recipes from large online datasets is time-consuming.
This project develops an **AI-based Recommendation System** that automatically suggests recipes according to nutritional composition or user goals (e.g., *"high protein, low carb under 600 kcal"*).

The model uses **content-based filtering** to compute similarity between recipes based on their macronutrient profiles (protein, carbohydrates, fats, and total calories). It also supports **goal-driven recommendations**, allowing users to receive personalized recipe lists that fit dietary preferences.

---

2. Objectives

- To design a **content-based recommendation system** for recipes.
- To use **nutrient composition** as the main similarity feature.
- To implement **goal-based filtering** (e.g., weight loss, keto, heart-friendly).
- To demonstrate **scalable and interpretable recommendations**.
- To preprocess, clean, and standardize multiple recipe datasets.

---

3. Dataset

An open-source Kaggle Dataset was used with the name: **Diets, Recipes And Their Nutrients**

Link: Diets, Recipes And Their Nutrients

Six CSV datasets were used, each containing recipes of different dietary categories:

Dataset File	Description
All_Diets.csv	Combined dataset with all recipes
dash.csv	DASH diet recipes
keto.csv	Ketogenic diet recipes

Dataset File	Description
mediterranean.csv	Mediterranean diet recipes
paleo.csv	Paleo diet recipes
vegan.csv	Vegan diet recipes

Total rows (before cleaning): 15,612
After deduplication: 7,126 unique recipes

Each file included the following columns:

- `Diet_type`
- `Recipe_name`
- `Cuisine_type`
- `Protein(g)`
- `Carbs(g)`
- `Fat(g)`
- `Extraction_day`
- `Extraction_time`

---

4. Data Preprocessing and Cleaning

Several cleaning and normalization steps were performed using **Python (pandas + PySpark)**:

1. **Column normalization:**
   - Fixed inconsistent names (e.g., “Carbohydrate_g” → “Carbs(g)”).
   - Standardized column cases and removed unwanted characters.
2. **Type conversions:**
   - Converted protein, carbs, and fat columns to numeric (`float`).
   - Removed non-numeric symbols like “g” or “gm”.
3. **Deduplication:**
   - Combined all six datasets into one.
   - Created a composite key of `Diet_type + Recipe_name` to remove duplicates.
4. **Missing values:**
   - Checked all columns — no missing data remained.
5. **Outlier scan:**
   - Identified extreme nutrient values using interquartile range (IQR) and P99 thresholds.
6. **Final clean dataset:**
   - 7,126 recipes × 9 columns.

---

5. Feature Engineering

The system converts each recipe into a **numerical feature vector** that represents its nutritional composition.

Steps:

1. Calculate **calories (kcal)** using

$$\text{kcal} = 4 \times \text{Protein(g)} + 4 \times \text{Carbs(g)} + 9 \times \text{Fat(g)}$$

2. Derive **macro-nutrient percentage composition**:

$$\begin{aligned}\text{Protein\%} &= \frac{4 \times \text{Protein(g)}}{\text{kcal}} \times 100 \\ \text{Carbs\%} &= \frac{4 \times \text{Carbs(g)}}{\text{kcal}} \times 100 \\ \text{Fat\%} &= \frac{9 \times \text{Fat(g)}}{\text{kcal}} \times 100\end{aligned}$$

3. Apply log transformation on calories (`log1p(kcal)`) to reduce scale imbalance.
4. Assemble all four features:
   `Protein%`, `Carbs%`, `Fat%`, and `log(kcal)`.
5. Use **StandardScaler** and **L2 Normalizer** to standardize vectors.

---

6. Recommendation Model Design

6.1 Content-Based Similarity Engine

Each recipe vector is compared to others using **cosine similarity**, which measures the angle between two nutrient vectors:

$$\text{similarity} = \frac{A \cdot B}{|| A || \quad || B ||}$$

When a user searches for a recipe (e.g., “Keto Cheesecake”), the system:

1. Finds its vector representation.
2. Calculates cosine similarity with every other recipe.
3. Returns the **Top-K most similar recipes** (same or other cuisines).

6.2 Goal-Based Recommendation

Users can also define nutritional goals such as:

- *High protein, low carb, under 600 kcal*
- *Keto, high fat, moderate protein*
- *Heart-friendly (low fat, moderate carb)*

The system:

1. Builds a **target vector** from user-specified macronutrient ranges.
2. Calculates cosine similarity between each recipe and the goal vector.
3. Returns recipes sorted by similarity score.

This method functions as a **goal-driven content-based recommender**.

---

7. Implementation Environment

- **Language:** Python 3.12
- **Platform:** Google Colab
- **Libraries Used:**
  - `pandas`, `numpy`
  - `matplotlib`, `seaborn` (for EDA)
  - `pyspark` (for scalable operations)
  - `sklearn` (used indirectly through PySpark MLlib)
- **Storage Format:** Parquet (for fast reload)

---

8. Exploratory Data Analysis (EDA)

Key findings from EDA:

Diet Type	Avg Protein (g)	Avg Carbs (g)	Avg Fat (g)	% Composition (Protein/Carbs/Fat)
DASH	69	160	100	21% / 49% / 30%
Keto	101	58	152	32% / 19% / 49%
Mediterranean	103	154	105	28% / 42% / 29%
Paleo	89	128	135	25% / 36% / 38%

Diet Type	Avg Protein (g)	Avg Carbs (g)	Avg Fat (g)	% Composition (Protein/Carbs/Fat)
Vegan	57	254	103	14% / 61% / 25%

Top cuisines observed per diet:

- DASH → American, Mediterranean
- Keto → American, Italian
- Mediterranean → Mediterranean, Italian
- Paleo → American, French
- Vegan → British, Italian

---

9. Example Results

(a) Similar Recipe Recommendation

Input: `"Keto Cheesecake"`
Output (Top 5 most similar):

1. Keto No-Bake Orange Creamsicle Cheesecake
2. Broccoli Cheddar Soup (Keto)
3. Keto Lemon Poppyseed Cheesecake
4. Keto Fat Bombs with Cacao and Cashew
5. Keto Strawberry Mousse Cake

(b) Goal-Based Recommendation

Input: *High protein, low carb, under 600 kcal*
Output:

Recipe	Diet	kcal	Protein%	Carb%	Fat%
Paleo Steak and Veggies	Paleo	268	36.5	19.7	43.9
Chop-Chop Beef Stir Fry	DASH	438	31.6	23.0	45.4
Keto Bacon & Shrimp Risotto	Keto	391	35.6	20.9	43.6
Paleo Protein Truffles	Paleo	326	36.2	14.9	48.9

---

10. Evaluation and Discussion

- The system performs **precise nutrient-based matching** using cosine similarity.
- Results are **interpretable** — users can see exact macronutrient ratios.
- Since the features are standardized, the recommendations remain robust across different diets and cuisines.
- The project demonstrates how content-based approaches can personalize healthy-eating suggestions without explicit user ratings.

---

11. Limitations

- Does not use collaborative filtering or deep learning embeddings.
- Assumes nutrition information is accurate for all recipes.
- Does not currently account for serving sizes or ingredients list.

---

12. Future Enhancements

- Include user-rating data to build **hybrid recommenders**.
- Add **ingredient-based similarity** using NLP.
- Include **visual interface for user input (React / Streamlit)**.
- Extend database with new cuisines and custom health goals.

---

13. Conclusion

This project successfully implements a **content-based recommendation system** for diet and recipe suggestions.
It analyzes and compares over 7,000 recipes based on their macronutrient profiles and calorie content, generating personalized suggestions that match specific dietary goals.
The model demonstrates that simple cosine-similarity techniques can produce meaningful and interpretable nutrition-based recommendations, forming a foundation for future hybrid recommender systems.