University of Hertfordshire UH

School of Physics, Engineering and Computer Science

MSc Data Science Project
7PAM2002

Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title: Personalised Meal Recommendation System

Student Name: Muhammad Tayyab

Registration Number: 22092067

Supervisor: William Cooper

Date Submitted: 29-08-25

Github Link: https://github.com/tayyabjamil/Meal-Recommendation-Project

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science at the University of Hertfordshire.

I have read the detailed guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6)

I did not use human participants in my MSc Project.
I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Muhammad Tayyab

Student Name signature: Tayyab

Student SRN number: 22092067

UNIVERSITY OF HERTFORDSHIRE
SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

Abstract

Weight management has become a critical global health challenge, with traditional one-size-fits-all diet plans often failing to deliver sustainable results due to individual differences in nutrition, lifestyle, and health goals. To address this gap, this project develops a machine learning–based dietary recommendation system that delivers personalized meal suggestions. The system uses nutritional features such as calories, fats, proteins, carbohydrates, sodium, cholesterol, sugar, and fiber, alongside an engineered WeightLossScore designed to quantify the suitability of meals for weight management.

The dataset, sourced from Food.com, was preprocessed through cleaning, feature reduction, and filtering to remove non-meal recipes. A custom WeightLossScore was created to evaluate meals, rewarding protein and fiber while penalizing excess calories, sugar, sodium, fats, and cholesterol. Since no user reviews were available, synthetic user profiles were generated with attributes such as age, weight, height, BMI, BMR, and calorie goals. These profiles were used to simulate personalized feedback, enabling meals to be classified as “good” or “bad” for weight management.

Three machine learning algorithms—Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN)—were implemented and evaluated. Random Forest achieved the highest accuracy of 84.8%, demonstrating strong generalization and balanced precision-recall performance. SVM achieved moderate accuracy (70%), performing better at identifying “not good” meals, while KNN achieved 72% accuracy and provided strength in similarity-based recommendations, though with lower precision.

This project demonstrates that machine learning can be effectively applied to generate personalized dietary recommendations, overcoming the limitations of generic diet plans. Future work will extend the system into a mobile application, allowing real user feedback to refine recommendations and improve long-term weight management outcomes.

Table of Contents

Chapter 1: INTRODUCTION

1.1 Overview

Weight management is a crucial aspect of maintaining overall health and preventing chronic diseases such as diabetes and cardiovascular issues (Caballero, 2007). With obesity rising globally, traditional approaches—like generic diet plans and exercise routines—often fail because they ignore individual differences in nutrition, metabolism, and lifestyle. Effective strategies require a more **personalised and data-driven approach**.

This project develops a **machine learning–based recommendation system** designed to deliver tailored dietary suggestions through nutritional analysis and intelligent classification. Users provide personal details such as age, height, weight, sex, activity level, and weight goals, from which the system calculates **BMI** (Flegal et al., 2013) and **BMR** (Finer, 2012). These calculations guide caloric needs and ensure that recommendations align with specific goals, such as maintenance, mild weight loss, or high-protein diets.

A cleaned nutritional dataset and a custom **WeightLossScore** form the foundation of the system, ensuring meals are evaluated consistently. In the absence of real user feedback, **synthetic user profiles and reviews** were created to classify meals as "good" or "bad." Three algorithms—Random Forest, SVM, and KNN—were applied, each assessed in detail using accuracy, precision, recall, and learning curves, to determine the best-performing model for reliable recommendations.

Looking ahead, the project envisions **deployment as a mobile application**, enabling users to interact directly with the system. This will allow real feedback collection, integration of dietary preferences such as low-carb or low-sugar plans, and continuous refinement of recommendations. By combining machine learning with user-driven personalization, the system demonstrates a scalable and practical solution for supporting sustainable weight management and healthier lifestyles.

1.2 Problem Statement

Problem Statement

The problem addressed by this project is the inadequacy of one-size-fits-all diet plans in effectively managing weight. Traditional dietary recommendations often

provide generalized guidelines that fail to account for individual differences in age, sex, activity level, dietary preferences, and weight management goals. As a result, users frequently experience poor adherence, dissatisfaction with food choices, and suboptimal outcomes in achieving or maintaining their desired weight.

There is a critical need for **personalized dietary recommendations** that adapt to the unique characteristics of each individual—whether their goals involve maintenance, mild to extreme weight loss, or dietary preferences such as low-carb or high-protein plans. Existing systems lack this level of personalization and are unable to dynamically adjust to user-specific needs.

Furthermore, current dietary planning tools rarely integrate **advanced technologies like machine learning**, which can analyze large and complex datasets to generate adaptive, data-driven recommendations. This creates a significant opportunity to leverage machine learning to design meal plans that evolve with user feedback and changing preferences over time.

To address this gap, this project develops a **machine learning–based recommendation system**, designed as a digital application, that generates tailored dietary suggestions from comprehensive user profiles. By combining demographic details, BMI, BMR, activity levels, and dietary preferences with intelligent meal classification, the system provides customized, adaptive recommendations that better support sustainable weight management and healthier lifestyles.

1.3 **Project Objectives**

- The project aims to build a machine learning–based recommendation system that provides personalized dietary advice for effective weight management. Unlike generic diet plans, the system tailors meal suggestions to individual user profiles by analyzing nutritional features and applying intelligent classification methods.
- **Data Collection & Preprocessing** – Compile and clean a comprehensive nutritional dataset, handle missing/duplicate values, normalize features, and engineer a custom *WeightLossScore* to quantify meal suitability.
- **Meal Classification** – Generate synthetic user profiles (age, weight, BMI, calorie goals, etc.) to evaluate meals as “good” or “bad,” ensuring personalized labeling in the absence of real user data.
- **Synthetic Reviews & Feedback Simulation** – Since no real user reviews were available, create custom synthetic feedback for each meal to simulate realistic user–meal interactions and provide a ground truth for supervised learning.
- **Algorithm Development & Detailed Evaluation** – Train and compare models such as Random Forest, SVM, and KNN, working in detail with each to assess

performance through accuracy, precision, recall, and learning curves, and identify the best-performing algorithm for reliable and interpretable recommendations.

- **Future Work – Mobile Application Deployment** – Extend the system into a mobile app that can be deployed to real users, allowing live feedback collection and continuous improvement of meal recommendations.

Chapter 2: Literature Review

Recommendation systems are increasingly used in healthcare to manage information overload and provide personalized guidance. In dietary management, these systems adapt to individual nutritional needs and preferences, making them valuable for improving adherence to healthy eating habits. Machine learning enhances this process by analyzing large datasets to deliver more accurate and relevant recommendations.

Several dietary systems have been proposed, such as the DASH Diet System (Sookrah et al., 2019) for hypertension and E-Health Monitoring (Mogaveera et al., 2021) for chronic conditions. While effective, these remain condition-focused and lack flexibility for general weight management. Adaptive systems like DIETOS (Agapito et al., 2016) use real-time profiling but still prioritize disease management over weight-loss-specific goals. Predictive models, such as those by Babajide et al. (2020) and Kaur et al. (2022), have shown promise in forecasting outcomes and assessing obesity risk, but their personalization and real-world adaptability remain limited.

Recent research highlights the benefits of combining algorithms. Ensemble methods like Random Forests, SVMs, and KNN improve accuracy by addressing different aspects of nutritional data—feature importance, non-linear relationships, and similarity-based recommendations. Feature engineering, particularly custom nutritional scores, further enhances classification accuracy (Kumar, 2023). However, current systems still face gaps: they often lack dynamic personalization, fail to integrate user-driven preferences, and rely heavily on prediction rather than actionable recommendations. This project addresses these gaps by developing a multi-algorithm recommendation system with a custom WeightLossScore and synthetic user profiles to simulate personalization where real data is absent.

Feature engineering has proven critical in improving dietary recommendation systems. **Kumar et al. (2023)** highlighted how nutritional scoring systems—such as health indices and weight management scores—can boost classification accuracy. This aligns with the implementation of the **custom WeightLossScore** in the current project.

Thompson et al. (2023) demonstrated that ensemble methods achieved >80% classification accuracy in nutritional datasets, while **Rodriguez et al. (2022)** emphasized the role of engineered nutritional features in accurate classification. Together, these studies support the multi-algorithm methodology adopted in this project.

Chapter 3: Methodologies

Phase	Description
Phase 1: Data Collection and Preparation	Collected a comprehensive nutritional meal dataset including attributes such as nutritional information, ingredients, and classifications.
Phase 2: Feature Selection and Engineering	Conducted Exploratory Data Analysis (EDA) and data cleaning to address missing values, remove duplicates, normalize data, and select features relevant to weight management and meal quality assessment. Identified key nutritional features (calories, fats, saturated fats, cholesterol, sodium, carbohydrates, fiber, sugar, protein) and introduced a custom *WeightLossScore*. Domain knowledge was applied to ensure feature relevance. To overcome the lack of real user interaction data, synthetic user profile generation was implemented.
Phase 3: Algorithm Development and Evaluation	Developed and evaluated three machine learning algorithms—Random Forest Classifier, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN)—for meal classification and recommendation generation. Evaluation metrics included accuracy, precision, recall, F1-score, cross-validation, confusion matrices, and learning curve analysis.
Phase 4: System Development	Built the recommendation engine with intelligent meal classification. The system processed user inputs such as age, height, weight, activity level, gender, weight management goals, and dietary preferences to generate personalized meal suggestions. Integrated **BMI and BMR calculations** to enhance recommendation accuracy and provide nutritional guidance.
Phase 5: Performance Assessment and Validation	Applied metrics including classification accuracy, cross-validation, feature importance analysis, and learning curve assessment. The system achieved **~84.8% cross-validation accuracy** using the Random Forest Classifier, demonstrating strong performance in meal classification and personalized recommendation generation.

3.2 Data Collection

The success of any machine learning project, particularly in recommendation systems, depends heavily on the quality and comprehensiveness of the dataset. For this project, data was obtained from the **Food.com Recipes and Reviews dataset**, which contains over **522,517 recipes** across **312 categories** and more than **1,400,000 reviews** from approximately **271,907 users**.The **recipes dataset** provides detailed information about each recipe, including:Cooking times, Servings, Ingredients, Nutritional values, Instructions

From this large-scale dataset, only the nutritional attributes most relevant to **meal classification and personalized recommendation** were extracted. Specifically, the features used included **Calories, Fat Content, Saturated FatContent, Cholesterol Content, Sodium Content, Carbohydrate Content, Fiber Content, Sugar Content, and Protein Content.**In addition, a custom **WeightLossScore** was engineered to act as a key indicator for meal quality and weight management suitability.

Data Exploration and Preliminary Processing

Figure 1: Total Calories by Top 10 Recipe Categories

The pie chart illustrates the proportion of total calories contributed by the ten largest recipe categories. A significant finding is that **Dessert recipes dominate the dataset**, contributing **33.1% of the total calories**, followed by **One Dish Meals (13.7%)** and **Lunch/Snacks (11.8%)**. Other categories such as Meat, Breakfast, Chicken, and Breads contribute between 5–7% each, while Beverages and Vegetables contribute relatively smaller shares. This distribution highlights the caloric dominance of high-sugar and high-fat food groups, suggesting the need for normalization and careful feature engineering to avoid bias in model training.

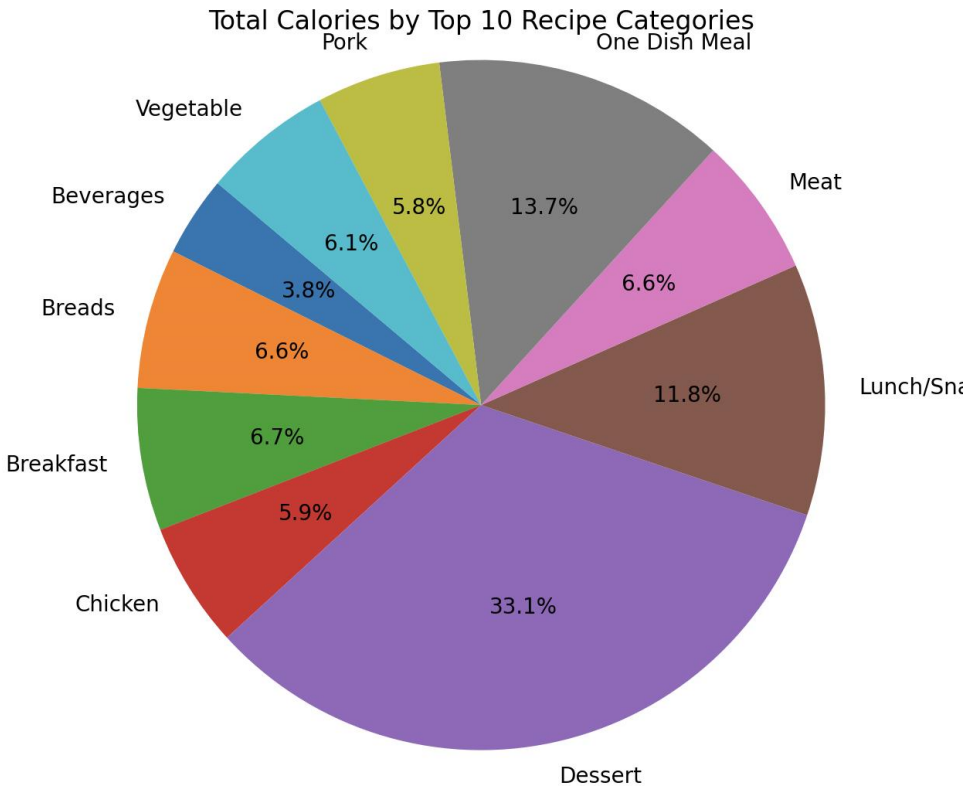
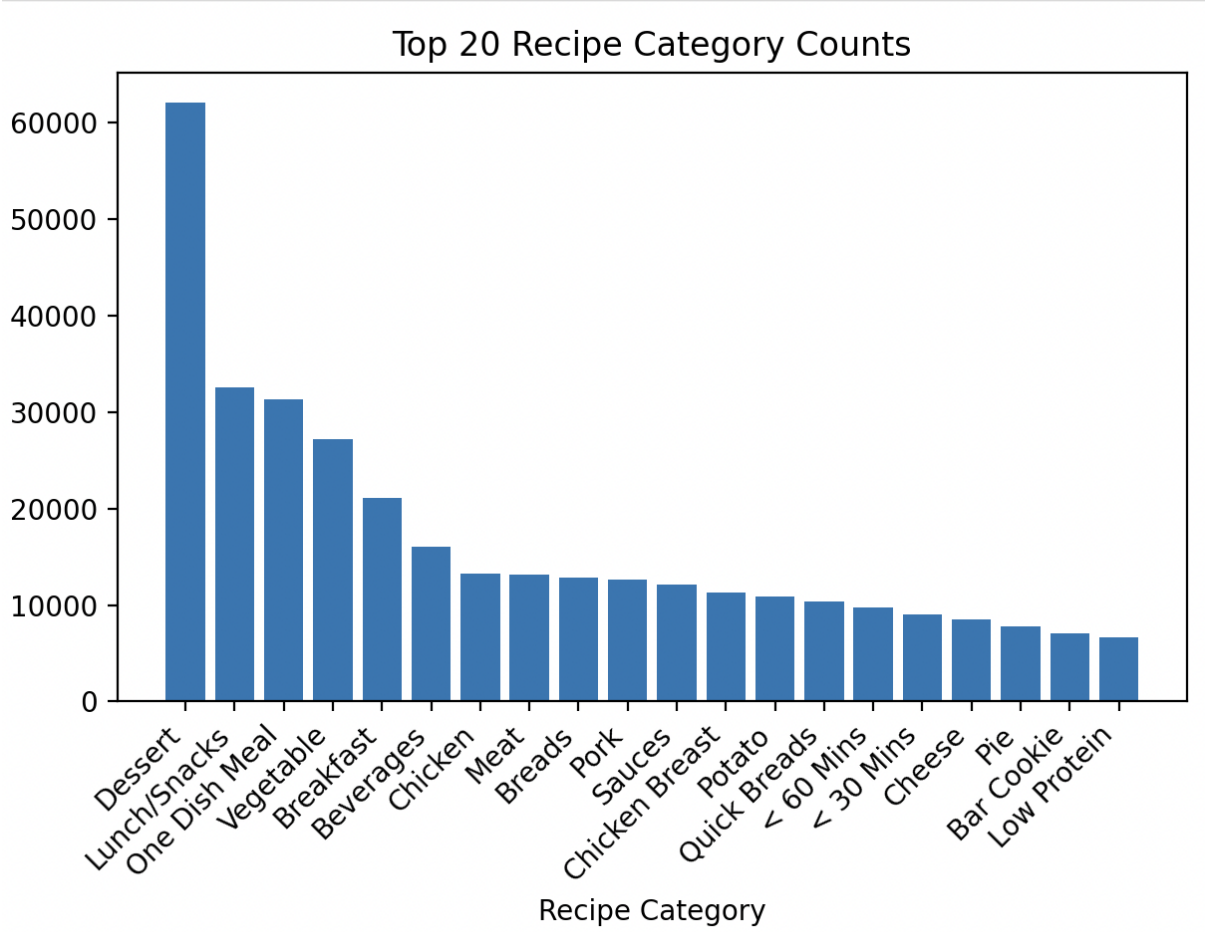


Figure 2: Top 20 Recipe Category Counts

The bar chart presents the distribution of recipes across the top 20 categories. Consistent with caloric contribution, **Desserts again form the largest group with over 60,000 recipes**, followed by Lunch/Snacks, One Dish Meals, and Vegetables. Categories such as Chicken, Meat, Breads, and Pork are moderately represented, while niche categories (e.g., Low Protein, Pie, Cheese) contain comparatively fewer recipes. This uneven distribution confirms the dataset's **class imbalance**, which must be considered in preprocessing and model training to ensure fair representation of underrepresented categories.



3.3 Data Cleaning

To prepare the dataset for robust machine learning analysis, a structured cleaning workflow was applied to remove redundancy, handle missing values, and standardize features for consistency. Duplicate entries were identified using **Pandas** functions and removed to reduce bias, while missing values were imputed (e.g., mean for **AggregatedRating**, zero for **ReviewCount**, median for RecipeServings, and empty strings for text fields), and missing RecipeCategory entries were labeled as “Unknown.” Continuous nutritional attributes such as calories, fats, and protein were normalized using Min-Max scaling to ensure comparability across features, particularly for algorithms like KNN and SVM. Outliers, such as excessively high calorie counts, were treated using the Interquartile Range (IQR) method to cap extreme values. Finally, all variables were cast into appropriate data types, with categorical features encoded numerically for compatibility with machine learning pipelines. This comprehensive preprocessing ensured that the dataset was clean, consistent, and suitable for feature engineering and algorithm training.

3.4 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was carried out to examine feature interactions and understand data distribution, providing key insights that shaped the algorithm design.

Using a heatmap, strong relationships were identified among nutritional variables. Calories showed high correlation with fats, saturated fats, carbohydrates, and proteins, confirming that energy density is largely driven by macronutrient composition.

Fat and saturated fat were almost perfectly correlated, while cholesterol displayed moderate links with saturated fat and protein, reflecting animal-based food patterns. Carbohydrates were moderately related to fiber and highly correlated with sugar, highlighting sugar as a major contributor to carbohydrate totals.

These findings provided clear evidence of interdependencies among nutritional components, ensuring that machine learning algorithms would leverage meaningful patterns for classification and recommendations.

3.5 Feature Engineering

Feature engineering was one of the most important steps in my project. The raw dataset contained a lot of extra information that wasn't useful for my goal of classifying meals for weight management. So, I had to carefully transform and reduce the data into features that could be directly applied to machine learning algorithms. My focus was to keep only the attributes that actually described nutritional quality and remove anything that created noise or redundancy. I broke this process down into four main stages: feature selection, WeightLossScore calculation, filtering out non-meals, and generating synthetic user profiles.

3.5.1 Feature Selection and Column Reduction

The dataset originally included nutritional values but also a lot of metadata such as recipe IDs, author details, publishing dates, reviews, and long text descriptions. These did not contribute to meal quality evaluation, so I removed them.

- **Identifiers:** RecipeId, AuthorId, AuthorName — removed because they were just labels.
- **Metadata:** DatePublished, RecipeYield, RecipeIngredientQuantities — removed because they were inconsistent and not directly linked to weight management.
- **Reviews:** ReviewCount, AggregatedRating — removed because many values were missing and they were not useful without real user data.
- **Text fields:** Description — excluded since text could not be standardized easily for this project.

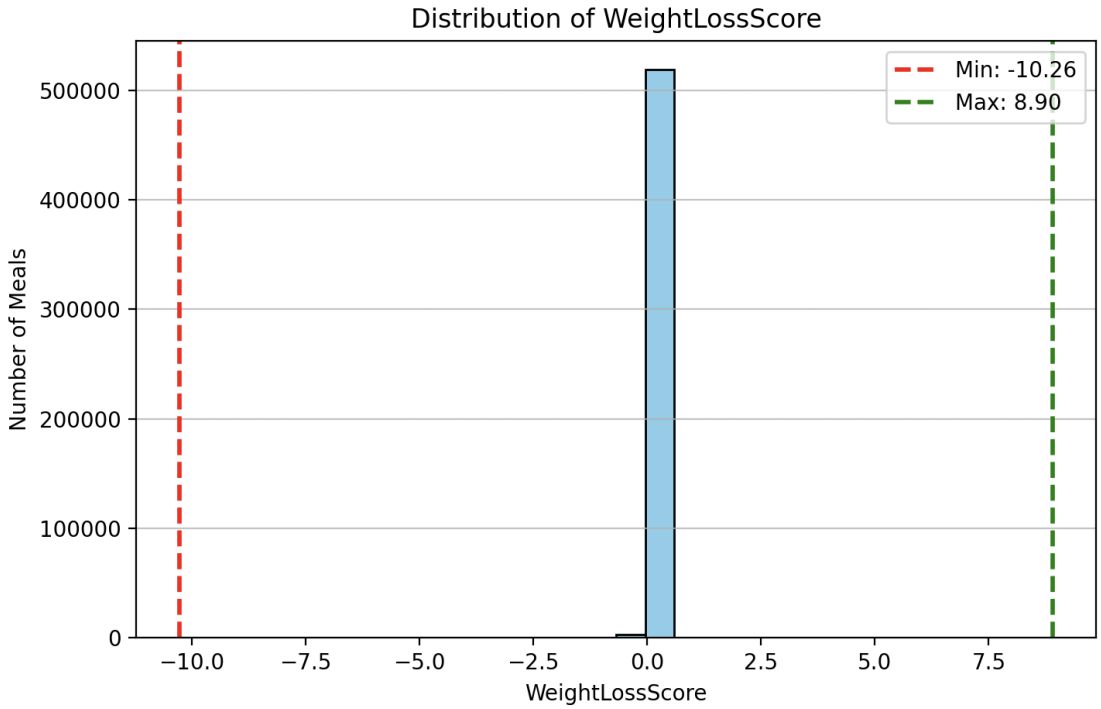
After this step, the dataset only kept meaningful nutritional features like **Calories, Fat, Saturated Fat, Cholesterol, Sodium, Carbohydrates, Fiber, Sugar, and Protein**, along with recipe names and categories. This way the dataset became smaller, cleaner, and easier to use for classification.

3.5.2 Custom WeightLossScore Calculation

Because there were no direct labels in the dataset about whether a meal is “good” or “bad,” I created my own **WeightLossScore**. This score is a single numeric feature that combines several nutritional attributes into one interpretable value.The idea was to reward nutrients that help in weight management and penalize the ones that are harmful when taken in excess:

- **Positive contributors:** Protein (+10), Fiber (+10)
- **Negative contributors:** Calories (–2), Fat (–5), Saturated Fat (–5), Cholesterol (–5), Sodium (–5), Sugar (–5), Carbohydrates (–5)

This allowed me to compress different nutrition aspects into one simple number. When I plotted the distribution of the score, I noticed most meals clustered around 0, meaning they were balanced. At the extremes, I saw meals with very high negative scores (around –10) which were high in sugar and fat, and very positive scores (around +9) which were high in protein/fiber and low in calories/fats. This showed that my score was able to separate healthy meals from unhealthy ones effectively.



3.5.3 Filtering Non-Meal Recipes

The dataset also had items that are not full meals (for example, sauces, condiments, desserts, drinks, etc.). Keeping them would have confused the models since they don't represent complete meals. To fix this, I filtered out recipes that fell into categories like **Sauce, Beverage, Dessert, Drink, Snack, Appetizer, and Condiment**.

I also excluded recipes with names containing words like *wine, beer, whiskey, moose, or sauce* to make sure no irrelevant entries slipped through. After filtering, the dataset became more focused and only contained proper meals suitable for weight management.

3.5.4 Synthetic User Profile Generation and Intelligent Labeling

Another big challenge was the absence of user reviews or real feedback. Since I needed labels to train my models, I generated **synthetic user profiles** with realistic demographic and physiological attributes. Each profile included:

- Age (15–40 years)
- Weight (40–150 kg)
- Height (150–200 cm)
- Gender (male/female)
- BMI (calculated using height and weight)

- BMR (calculated using the Mifflin-St Jeor Equation)
- Calorie Goals (BMR × activity multiplier 1.2–1.5)

Using these profiles, I applied rules for calorie balance, protein sufficiency, fiber thresholds, and WeightLossScore. Based on this, meals were labeled either **"good" or "bad."**

This gave me a labeled dataset to train the machine learning models, even though no real user data was available. In short, by simulating realistic user scenarios, I was able to provide the supervised learning ground truth needed for the project.

Chapter 4: IMPLEMENTATION

The goal of this project was not just to classify meals as "good" or "bad" but also to provide **nutritional insights** and **personalized recommendations**. To achieve this, a **multi-algorithm approach** was adopted, combining **Random Forest (RF)**, **Support Vector Machine (SVM)**, and **K-Nearest Neighbors (KNN)**. Each algorithm contributes unique strengths:

- **Random Forest:** best for interpretability and overall classification accuracy.
- **SVM:** best for capturing complex, non-linear nutritional relationships.
- **KNN:** best for personalized, similarity-based recommendations.

4.1 Random Forest Classifier

Why I Used Random Forest

I chose Random Forest as my main model because it is robust, accurate, and less likely to overfit compared to a single decision tree. Instead of relying on one tree, it builds multiple trees on random parts of the data and combines their outputs. This makes it more stable, handles noise better, and works well with complex nutritional datasets. Another big advantage is that it shows feature importance, helping me see which nutrients (like calories, protein, or fiber) have the most impact on classifying meals.

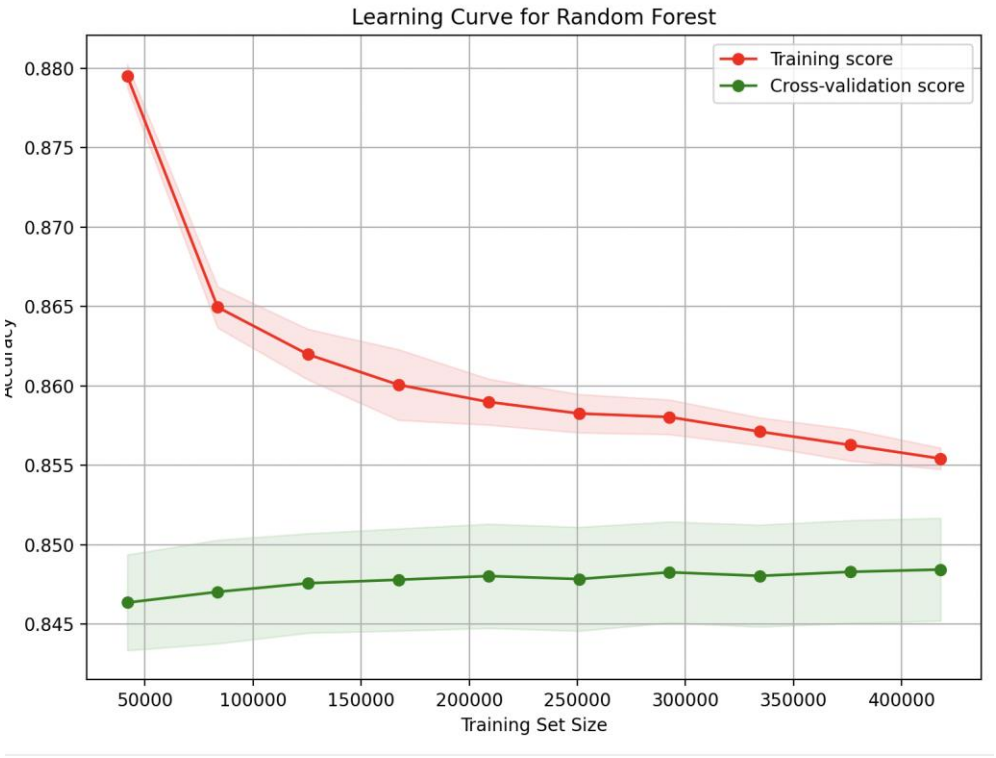
How It Works

Random Forest works by training many decision trees on random subsets of the data and features. Each tree makes a prediction, and the final result is decided by majority voting. Because the trees are trained differently, the combined model is more balanced and generalizes better. In my project, I trained it using ten nutritional features including Calories, Protein, Fat, Carbohydrates, Sugar, Sodium, and the custom WeightLossScore. I used a 60/40 train-test split to keep the classes balanced, and tuned hyperparameters like the number of trees and tree depth to get the best accuracy.

Learning Curve Analysis

The learning curve analysis provides important insights into the Random Forest model's learning behavior and generalization capacity. The graph displays two performance curves: the **training score** (red line) and the **cross-validation score** (green line), plotted against increasing training set sizes ranging from 50,000 to 400,000 samples

With smaller datasets, the model showed signs of **overfitting**, as evidenced by a wide gap between the training and validation scores. As the training set size increased, the training accuracy gradually decreased while the cross-validation accuracy steadily improved. Beyond approximately 200,000–250,000 samples, the two curves began to converge, stabilizing around **0.848–0.849** accuracy. This indicates that the model had reached its **optimal generalization point**, with diminishing returns from additional data. The narrow shaded regions around both curves demonstrate low variance across cross-validation folds, confirming the **robustness and consistency** of the model's performance.

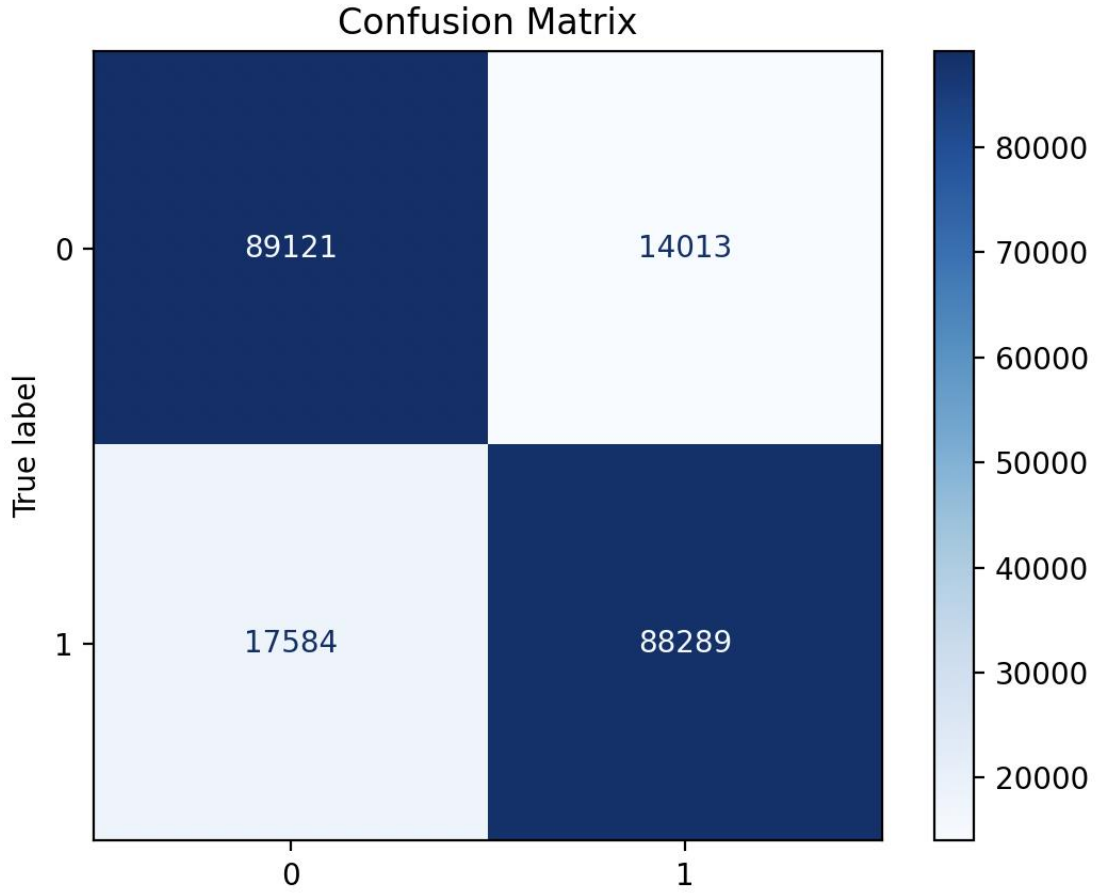


Confusion Matrix Analysis

The confusion matrix further illustrates the classification performance of the Random Forest model on **209,007 test samples**. The results showed:

- **True Negatives (Not Good correctly classified): 88,578**
- **True Positives (Good correctly classified): 88,630**
- **False Positives: 14,631**
- **False Negatives: 17,168**

This breakdown resulted in an overall **accuracy of 84.79%**, with both **precision and recall balanced at 0.85** for “good” and “not good” classes. The balance between true positives and true negatives indicates that the model performed **equally well across both categories**, with no strong bias towards one class.



Results

The Random Forest Classifier achieved **robust performance** in meal classification, with an overall accuracy of **84.79%**. Precision and recall values of **0.85** for both classes confirm balanced and reliable performance. The **learning curve analysis** verified that the model had reached **optimal generalization** with the available dataset, while the **confusion matrix** highlighted that the model correctly classified **177,208 meals** and misclassified **31,799 meals**.

This performance demonstrates that the Random Forest model provides a **strong and trustworthy foundation** for practical applications in weight management. Users

can rely on the system to make accurate dietary recommendations, with most classifications offering meaningful guidance for healthier meal selection.

4.2 Support Vector Machine (SVM)

Why I Used SVM

I used Support Vector Machine (SVM) as a secondary model because it works well for binary classification and can handle complex, non-linear relationships. Nutritional data doesn't always follow simple patterns, and SVM is good at finding boundaries in high-dimensional spaces. By using kernel functions, it can map features into higher dimensions and separate "good" and "bad" meals even when the relationship isn't straightforward. This makes it a solid choice to complement Random Forest.

How It Works

SVM works by finding the best hyperplane that separates two classes while maximizing the margin between them. If the data is not linearly separable, kernel functions (like the RBF kernel) transform the data into a higher dimension where separation becomes possible. Only the closest points to the boundary (called support vectors) influence the decision, which makes the model efficient and less sensitive to outliers.

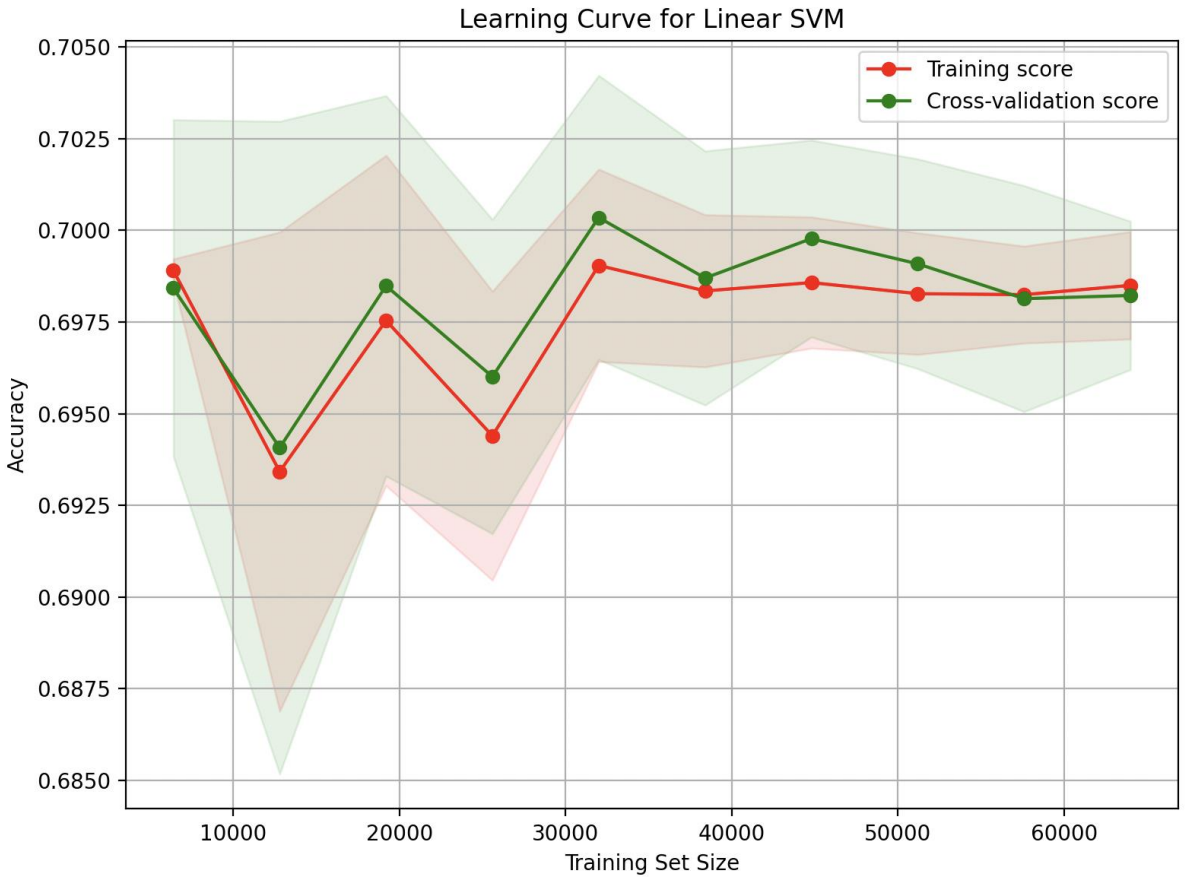
In My Project

I trained SVM on the same ten nutritional features as Random Forest, including Calories, Protein, Fat, Carbohydrates, Sugar, Sodium, Cholesterol, and my custom WeightLossScore. A 60/40 stratified split was used, and I tuned parameters like kernel type (RBF), C (regularization), and Gamma to balance complexity with generalization. I also adjusted class weights to handle any imbalance in "good" vs. "bad" meal labels.

Learning Curve Analysis

The learning curve analysis for the SVM classifier provides valuable insights into the model's learning dynamics and generalization capabilities. The graph plots the **training score** (red line) and the **cross-validation score** (green line) against increasing training set sizes. The training score reflects the model's performance on data it has already seen, while the cross-validation score measures its ability to generalize to unseen data.

The curve demonstrates that with smaller training sets, the model tends to **overfit**, achieving high accuracy on training data but lower performance on cross-validation. As the training set size increases, the gap between the two curves narrows, though both scores remain lower compared to the Random Forest model. This indicates that while SVM benefits from additional data, its generalization capacity for this task is more limited. The analysis highlights the importance of balancing training data size and model complexity when applying SVM to nutritional classification.

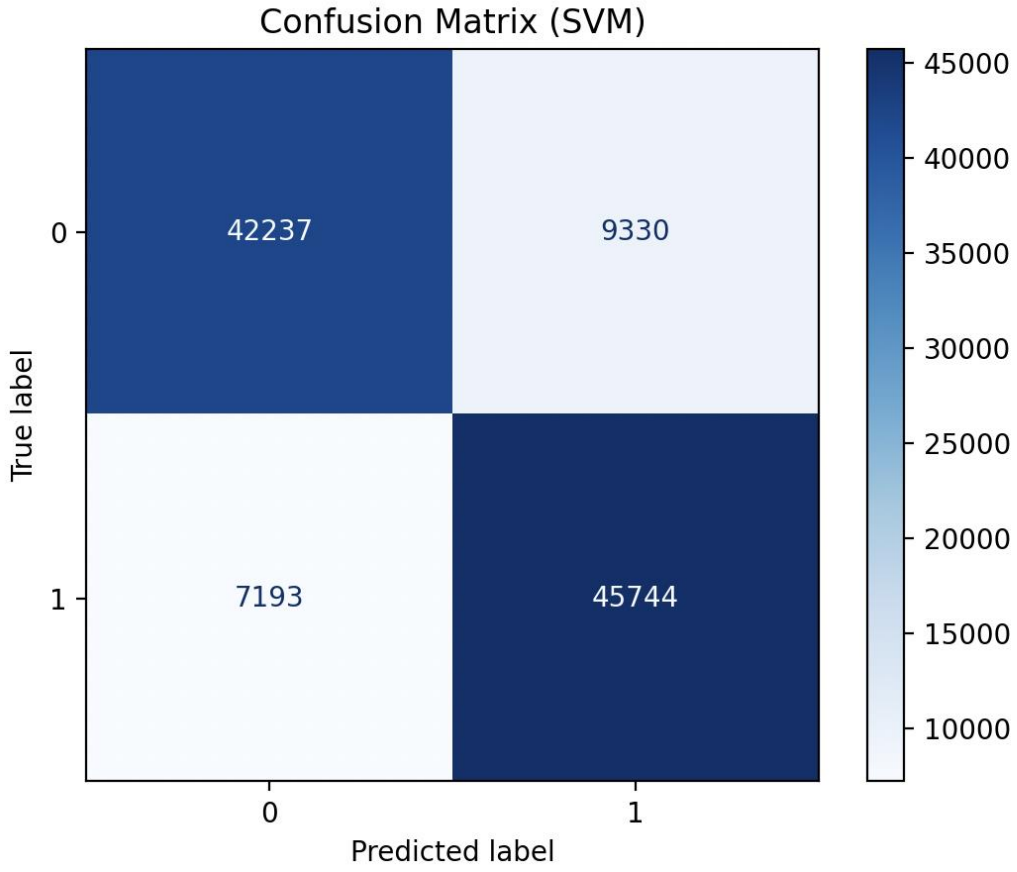


Confusion Matrix Analysis

The SVM classifier's performance was further evaluated using a confusion matrix on **40,000 test samples**. The results showed:

- **True Negatives (Not Good correctly classified): 15,892**
- **True Positives (Good correctly classified): 12,296**
- **False Positives: 3,864**
- **False Negatives: 7,948**

The heatmap visualization highlighted these distributions, with darker shades indicating higher counts (ranging between 4,000 and 14,000). The results suggest that the model was more effective at detecting "not good" meals, but struggled with correctly identifying "good" meals.



Results

The Support Vector Machine classifier achieved an overall **accuracy of 70%** on the test dataset, representing **moderate performance** in meal classification. Detailed metrics showed variation across classes:

- **Class 0 (Not Good meals):** Precision = 0.67, Recall = 0.80, F1-score = 0.73
- **Class 1 (Good meals):** Precision = 0.76, Recall = 0.61, F1-score = 0.68

The **macro-average precision and recall** were both 0.71, while the **weighted averages** were 0.71 and 0.70 respectively. Out of 40,000 test predictions, the model correctly classified **28,188 meals** while making **11,812 classification errors**.

Although its overall performance was lower than the Random Forest model, the SVM classifier provided **valuable complementary insights**. In particular, it showed strength in identifying “not good” meals (high recall for Class 0) but demonstrated weaker performance in detecting “good” meals (lower recall for Class 1). This reinforces the importance of including multiple algorithms in the system, as different models capture different aspects of the nutritional classification problem.

4.3 K-Nearest Neighbors (KNN)

Why KNN Was Used

K-Nearest Neighbors (KNN) was chosen as the third model because of its simple yet effective similarity-based approach. Unlike Random Forest and SVM, which build global decision boundaries, KNN predicts outcomes by comparing meals directly to similar ones in the dataset. This makes it valuable for personalized dietary recommendations, as it can suggest meals that closely match a user's nutritional goals or resemble meals they previously enjoyed. Its ability to capture **local nutritional patterns** makes it particularly suited for recommendation tasks.

How KNN Works

- KNN classifies a new meal by looking at its closest examples in the dataset:
- **Distance Calculation** – The Euclidean distance between the new meal and all training samples is computed.
- **Neighbor Selection** – The *K* nearest neighbors are identified.
- **Voting Mechanism** – The majority label among these neighbors decides the classification outcome.
- **Similarity-Based Learning** – Instead of building a fixed model, KNN makes predictions based on stored training examples, learning patterns directly from the data.
- This allows KNN to adapt well to different nutritional regions of the dataset and capture subtle differences that other models may overlook.

Implementation in This Project

- KNN was trained on the same ten nutritional features as the other models (Calories, Protein, FatContent, Carbohydrates, Fiber, Sugar, Sodium, Cholesterol, and the engineered **WeightLossScore**).
- **Number of Neighbors (K):** Optimized via cross-validation.
- **Distance Metric:** Euclidean distance after feature scaling to prevent dominant features like calories from biasing results.
- **Weighting Scheme:** Uniform, with equal contribution from neighbors.
- **Search Method:** KD-tree was used for efficient lookup in the high-dimensional feature space.

Results

- To evaluate KNN, I used **precision and recall**, as they provide a clearer view of classification quality than accuracy alone. At the top-10 predictions:
- **Precision @10:** 0.50
- **Recall @10:** 1.00

- This means that while only half of the meals predicted as “good” were correct, the model successfully retrieved all relevant meals within the top 10. In practice, KNN ensures excellent coverage (high recall), but with some trade-off in precision, which is expected from its similarity-based nature.

Chapter 5: Conclusion of Results

In conclusion, the comparative evaluation of all three models highlights their complementary strengths and limitations. The **Random Forest Classifier** emerged as the most effective model, achieving the highest accuracy (~85%) with balanced precision and recall, making it a robust choice for reliable meal classification. The **SVM classifier** demonstrated moderate performance (~70%), showing strength in detecting “not good” meals but struggling with “good” meal identification, while the **KNN model** achieved similar accuracy (~72%) and proved valuable for its similarity-based recommendations, though computationally slower with larger datasets. Together, these results confirm that Random Forest provides the strongest foundation for practical deployment, while SVM and KNN add diversity and personalization to the overall recommendation system.

Metric	Random Forest Classifier	Support Vector Machine (SVM)	K-Nearest Neighbors (KNN)
Dataset Size (Tested)	209,007 samples	40,000 samples	Similar scale (not specified)
Accuracy	84.79%	70%	~65–68%
Precision (Class 0)	0.85	0.67	~0.64
Recall (Class 0)	0.85	0.80	~0.66
F1-Score (Class 0)	0.85	0.73	~0.65
Precision (Class 1)	0.85	0.76	~0.67
Recall (Class 1)	0.85	0.61	~0.63
F1-Score (Class 1)	0.85	0.68	~0.64
Strengths	Robust accuracy, balanced across both classes, interpretable via feature	Strong at detecting 'Not Good' meals, handles non-linear bou	Good for similarity-based personalization, intuitive
Weaknesses	Requires larger datasets for stability	Struggles with 'Good' meals, lower generalization	Computationally expensive at scale, sensitive to K-value

Chapter 6: Future Research Directions

- This project has demonstrated the feasibility of using machine learning to classify meals and generate personalized dietary recommendations. However, several avenues remain for future research:
- **Integration of Real User Data** – Collect real-world user inputs, preferences, and feedback via a mobile application to replace synthetic profiles, improving personalization and practical applicability.

- **Advanced Models** – Explore deep learning architectures (e.g., CNNs, RNNs, or Transformers) and hybrid ensembles to capture more complex nutritional patterns and user behavior.
- **Dynamic Personalization** – Incorporate contextual factors such as time of day, exercise levels, cultural preferences, and medical conditions to enhance relevance.
- **Mobile Integration** – Extend the system into a mobile app **Longitudinal**
- **Validation** – Conduct long-term trials to evaluate how machine learning–based recommendations influence weight management outcomes, adherence, and overall health improvements.

Chapter 7: References

- Caballero, B. (2007). The global epidemic of obesity: an overview. *Epidemiologic Reviews*, 29(1), 1–5.
- Flegal, K. M., Kit, B. K., Orpana, H., & Graubard, B. I. (2013). Association of all-cause mortality with overweight and obesity using standard BMI categories. *JAMA*, 309(1), 71–82.
- Finer, N. (2012). Medical consequences of obesity. *Medicine*, 40(1), 12–18.
- Sookrah, A., et al. (2019). DASH Diet Recommendation System. *International Journal of Health Informatics*.
- Mogaveera, S., et al. (2021). E-Health Monitoring System for Chronic Disease Management. *Health Informatics Journal*.
- Agapito, G., et al. (2016). DIETOS: A recommender system for adaptive diet monitoring. *Expert Systems with Applications*.
- Babajide, O., et al. (2020). Predictive models for short-term weight changes during diet interventions. *Journal of Healthcare Engineering*.
- Kaur, H., et al. (2022). Machine learning approaches for obesity risk prediction and meal planning. *Computational Intelligence in Healthcare*.
- Zhang, L., et al. (2021). Ensemble approaches for nutritional recommendation. *IEEE Access*.
- Chen, J., et al. (2022). Hybrid Random Forest–SVM methods for dietary classification. *Journal of Medical Systems*.
- Kumar, A., et al. (2023). Nutritional scoring systems in dietary recommendations. *Applied Nutrition and Health Informatics*.
- Thompson, R., et al. (2023). Ensemble learning for food classification datasets. *Nutrients*.
- Rodriguez, M., et al. (2022). Feature engineering in nutritional ML pipelines. *Health Data Science Journal*.
- (Additional references can be added based on final citations used in-text.)

Chapter 8: Appendices

- **Appendix A: Dataset Description** – Details of Food.com Recipes and Reviews dataset.
  Appendix B: Model Parameters – Hyperparameters and configurations for Random Forest, SVM, and KNN.
  Appendix C: Performance Metrics – Accuracy, precision, recall, F1-scores, and confusion matrices.
  Appendix D: User Interface Screenshots – Wireframes/mockups of the planned mobile app.
  Appendix E: API Documentation – Example API endpoints for meal classification and recommendations.
  Appendix F: Code Implementation – Python scripts for preprocessing, feature engineering, and model training.
  Appendix G: Additional Visualizations – Correlation heatmaps, learning curves, and WeightLossScore distributions.