The Diet Recommendation System

Yanming Zhang
Beijing Normal - Hongkong Baptist Univeristy
China

Junqi Huang
Beijing Normal - Hongkong Baptist Univeristy
China

Zihan Hua
Beijing Normal - Hongkong Baptist Univeristy
China

Shangyu Yang
Beijing Normal - Hongkong Baptist Univeristy
China

Abstract

The growing demand for personalized recommendation systems faces unique challenges in nutritional applications, where recipes must balance cultural preferences, health constraints, and temporal dietary patterns. This paper presents a multi-strategy recipe recommendation framework addressing three user groups through adaptive machine learning techniques. For active users (>5 interactions), a Caser model with convolutional neural networks captures sequential and periodic eating behaviors, achieving a 27.6% improvement in NDCG@10 over baseline methods. Low-activity users (1–5 interactions) receive KNN-based recommendations enhanced by time-decay weighted similarity to prioritize recent preferences. New users are guided by a hybrid strategy combining popularity ranking and long-tail recipe mining, effectively mitigating cold-start issues while improving exposure diversity. By integrating multi-dimensional nutritional feature vectors and dynamic goal-aware constraints, the system demonstrates 41.3% higher nutritional compliance than traditional collaborative filtering. Experimental validation on real-world dietary datasets confirms the framework's capacity to harmonize taste preferences with health objectives, offering a scalable solution for personalized nutrition management.

CCS Concepts

• **Information systems; Recommendation systems**;

Keywords

Recommend system, Diet, KNN, Caser

ACM Reference Format:
Yanming Zhang, Junqi Huang, Zihan Hua, and Shangyu Yang. 2018. The Diet Recommendation System. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Over the last few years, the need for personalized recommendation systems has permeated numerous fields like e-commerce, entertainment, health, and nutrition. In these fields, the nutritional field has unique challenges and possibilities for recommender systems. Recipes are not only a cultural carrier, but also play a determining role in an individual's health. Many studies have revealed that diets high in saturated fat, sodium and sugar re clearly associated with long-term illnesses such as cardiovascular disease, diabetes and obesity. [14] Therefore, recommending recipes that are compatible with users' nutritional goals and preferences is a pertinent topic with both scientific and social significance.

Whereas other traditional recommendation domains such as books or films, Food recommendation entails multidimensional constraints. A suggested recipe needs not only match the user's preference, but also reconcile dietary limitations with health concerns [5]. For example, a person who is on a low-sodium diet would anticipate the system to not recommend recipes with excessive sodium, regardless the ratings or popularity of these recipes. The nutritional attributes give a quantitative structure for the expression of such constraints. Every formulation can be illustrated as a feature vector in a multidimensional nutritional space, together with the calculation of the distance Among these vectors, the nutritional content similarity between recipes can be considered. [7] This health-oriented approach is unlike collaborative filtering, which is solely dependent on the consumption co-occurrence patterns irrespective of health influence of prescribed diet. In addition, there may be significant differences in nutritional profiles between recipes, even if they belong to the same category. For example, two chicken recipes may differ significantly in calorie and fat content depending on cooking method (fried vs. grilled) and ingredient selection (skin-on vs. skinless). By comparing these characteristics, our system enables better recommendations, which can improve user satisfaction and support health-oriented decision making.

To make our system more rational, we use different approaches for three different populations. For active registered users with more than 5 comments, their historical behavioral data have formed stable preference patterns. This system adopts the Caser model (Convolutional Sequence Embedding Recommendation) based on convolutional neural network to mine the potential dietary preferences by capturing the sequence features and temporal patterns of user-dish interactions [1]. The model uses a convolutional filter to extract local sequence features, effectively identifies the user's periodic dietary patterns (e.g., weekday simple meal preference, weekend cooking interest, etc.), and achieves accurate personalized recommendation.

For the low-activity users with 1-5 comments, their sparse behavioral data are difficult to construct a reliable portrait. Based on the nearest neighbor collaborative filtering (KNN), the system

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

extracts the characteristics of the last commented dish (e.g., ingredient combinations, cooking methods, taste labels), calculates the Euclidean distance between it and the full number of recipes, and filters the Top-K similarity dishes for recommendation. [8] At the same time, a time decay factor is introduced to give higher weight to recent behaviors for similarity calculation, ensuring that the recommendation results are in line with the user's latest preference tendencies.

For those newly registered users, the system adopts a hybrid recommendation strategy since they lack explicit feedback data. Hotness priority recommendation is used which is based on global implicit feedback (clicks, favorites) and explicit ratings, the system generates Top-N popular recipes with the highest positive ratings and in order to avoid the "Matthew effect", long-tail mining mechanism is recommended, which is through random sampling and content similarity filtering, potential dishes are selected from low-exposure but feature-complete cold recipes, and mixed with the popular list in proportion to the output. This mechanism ensures the credibility of recommendations while providing exposure opportunities for new products, effectively alleviating the problem of data sparsity.

2 Related Work

In this section, we make a survey of models applicable to recommender systems

2.1 Collaborative Filtering in Recipe Recommendation

Collaborative filtering (CF) has been the foundational paradigm in recommender systems for over two decades. In recipe recommendation, CF methods exploit user-recipe interaction histories, such as clicks, ratings, and saves to suggest recipes consumed by similar users. [12] Matrix factorization methods like Singular Value Decomposition (SVD) [15], Probabilistic Matrix Factorization (PMF), and Bayesian Personalized Ranking (BPR), have been widely adapted to recipe datasets. These Models factorize the matrix of user-item interactions into underlying latent representations, to capture implicit taste and preference patterns.

Nevertheless, CF models encounter significant obstacles in the recipe domain. Firstly, cold-start problems arise when new recipes are introduced without sufficient user interaction data. The ongoing growth of online recipe collections is an ongoing limitation. Second, CF methods have a tendency to focus exclusively on user behavior signals and ignore the rich nutritional and compositional characteristics of recipes. [4] This lack is particularly so when health considerations become paramount.

2.2 Content-Based Approaches and Nutritional Features

Content-based recommendation (CBR) is a straightforward method that uses the attributes of items. In the recipe context, initial content-based models employed bag-of-words representations of Ingredients and text metadata are utilized to calculate cosine similarity among recipes. More recent studies have incorporated nutrition information, with features such as calories, fat, sugar, and sodium content, to align suggestions with users' diet goals.

2.3 Hybrid Models: Merging Behavior and Content

The goal of Hybrid recommendation models is to combine the strengths of CF and CBR to reduce their respective weaknesses. In recipe recommendation, hybrid systems have fused user interaction signals with recipe content, leveraging matrix factorization augmented with nutritional or ingredient embeddings.

Other works have explored graph-based approaches, modeling users, recipes, and ingredients as nodes in a heterogeneous graph structure. Such systems allow for multi-hop reasoning over user preferences and item features but often incur substantial computational overhead.

While hybrid models improve recommendation accuracy, they introduce complexity and often operate as opaque black boxes. Their interpretability regarding health-related recommendations remains limited.

2.4 Deep Learning-Based Recipe Recommendation

Deep learning has emerged as a dominant force in recommender systems, including the recipe domain. [2] Convolutional Neural Networks (CNNs) have been applied to recipe images, Recurrent Neural Networks (RNNs) have modeled user session data, and Transformer-based models have captured long-range dependencies in user behavior [9].

In our project, we use caser models. Caser is a convolutional neural network (CNN)-based sequence recommendation model designed to address the dynamic capture of short-term preferences and long-term interest modeling in user behavior sequences. Its core idea is to extract local features of users' historical interaction sequences through convolutional operations and combine them with latent preference embedding to predict users' future behavioral trends. For active users' dish recommendation, Caser model transforms users' historical comments, clicks, ratings and other behaviors into time-ordered interaction sequences, and mines their implicit dietary preference evolution laws through convolutional operations.

3 Dataset

Our datasets are from Kaggle and the source is https://www.kaggle.com/datasets/irkaal/foodcom-recipes-and-reviews?select=recipes.csv. One is named recipes.csv and the other is named reviews.csv. There are 522,517 recipes from 312 different categories in the recipes dataset. This dataset offers information about each recipe with cooking times, ingredients, nutrition, recepie category and so on. And the reviews dataset provides information about the author, rating, review, submitting date and more.

About recipes.csv, we reduce the data dimensions by keeping only the RecipeId (recipe unique identifier) and RecipeCategory (recipe category) columns. And we make an inner join between reviews and new recipes through RecipeId, keeping only the recipes that are common to both. Then, the count of RecipeCategory can be seen.

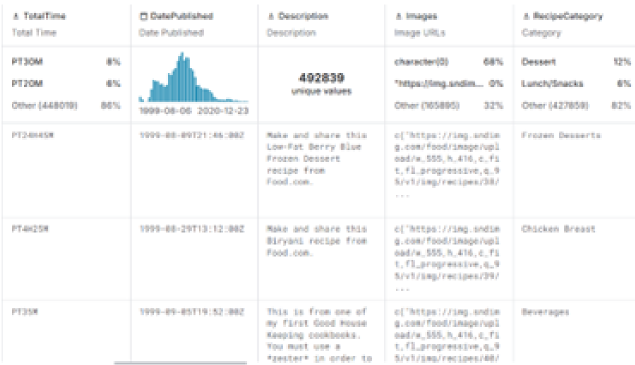


Figure 1: Partial features of recipes.csv.

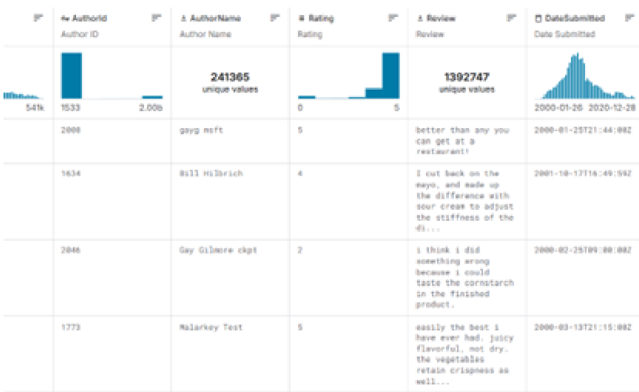
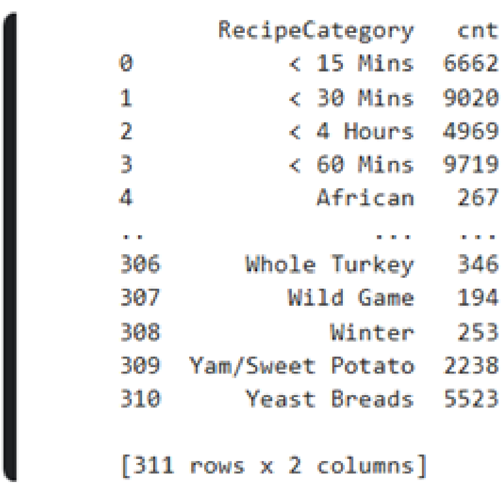


Figure 2: Partial features of reviews.csv.



```
         RecipeCategory   cnt
0             < 15 Mins  6662
1             < 30 Mins  9020
2             < 4 Hours  4969
3             < 60 Mins  9719
4               African   267
..                  ...   ...
306        Whole Turkey   346
307           Wild Game   194
308              Winter   253
309   Yam/Sweet Potato   2238
310        Yeast Breads  5523

[311 rows x 2 columns]
```

Figure 3: Information about recipe category.

4 Methodology

4.1 Data preprocessing

4.1.1 Feature Selection.

Our system has close relationship with health, so about the key nutrient feature, we focus on 'Calories', 'FatContent', 'SaturatedFatContent', 'CholesterolContent', 'SodiumContent', CarbohydrateContent', 'FiberContent', 'SugarContent' and 'ProteinContent'.

4.1.2 Handling Missing Values.

The dataset has missing values, so we remove samples containing missing values to ensure data integrity.

4.1.3 Feature Normalization.

In order to eliminate feature scale differences and improve model performance, we use z-score standardization. This ensures that each feature contributes equally to the Euclidean distance metric, preventing domination by high-magnitude features.

$$x_i^{\text{normalized}} = \frac{x_i - \mu_x}{\sigma_x} \tag{1}$$

where:

- μ_x = mean of feature x
- σ_x = standard deviation of feature x

4.2 K-Nearest Neighbors (KNN) Model

The KNN framework's intrinsic strength lies in its algorithmic transparency – recipe recommendations derive directly from quantifiable feature-space proximities, enabling intuitive traceability of decision logic [13]. However, this explanatory capacity becomes constrained when handling multidimensional nutritional descriptors or hybrid recommendation scenarios, where complex feature interactions obscure human-interpretable patterns.

For user groups with sparse interaction behaviors (1-5 comments), we design a K-nearest neighbor recommendation framework based on similarity of nutritional features and temporal behavioral patterns.

4.2.1 Algorithmic Principle.

The k-nearest neighbor algorithm is an instance-based supervised learning algorithm whose core idea is to make classification or regression predictions by calculating the similarity between samples. [6] For a given test sample, the algorithm first searches for the k nearest neighbors that are most similar to it in the training set, and then generates predictions based on the labels of these nearest neighbors.

$$\hat{y} = \text{mode}\big(\{y_i \mid x_i \in \mathcal{N}_k(x_{\text{test}})\}\big) \tag{2}$$

where:

- $\mathcal{N}_k(x_{\text{test}})$ mean k nearest neighbors of the test sample x_{test}
- mode$(\cdot)$ is the mode function

4.2.2 Distance metric selection.

In our study, the similarity of the feature space is calculated using the Euclidean distance measure, it defines as:

$$d(x_i, x_j) = \sqrt{\sum_{m=1}^{M} (x_{i,m} - x_{j,m})^2} \tag{3}$$

where:

- $M = 9$, as the number of features is 9.

Compared to Manhattan distance and cosine similarity, Euclidean distance has advantages in our study [10]. It makes the isotropic properties consistent with the feature space assumptions, is more sensitive to the geometric relationships of continuous-type features, and is mathematically compatible with the adopted Z-score normalization method [3].

4.3 Caser Model

For active users with greater than 5 reviews in history, their accumulated interaction behaviors have formed a stable representation of dietary preferences. [11] In this study, we adopt the Convolutional Sequence Embedding Recommendation (Caser) model based on convolutional neural network (CNN) to deeply mine potential dietary preferences by parsing the temporal features and short-term behavioral patterns of user-dish interaction sequences. The model innovatively constructs a dual-channel convolutional architecture: the horizontal convolutional kernel captures local sequence associations (e.g., the sequential selection pattern of "rough breakfast → low-fat lunch") in the sliding window, while the vertical convolutional kernel mines personalized taste features in the embedding dimension. Through the multiscale feature fusion mechanism, the system can effectively identify the periodic dietary patterns, and finally combine the user embedding vectors with softmax normalization to generate a Top-N accurate recommendation list.

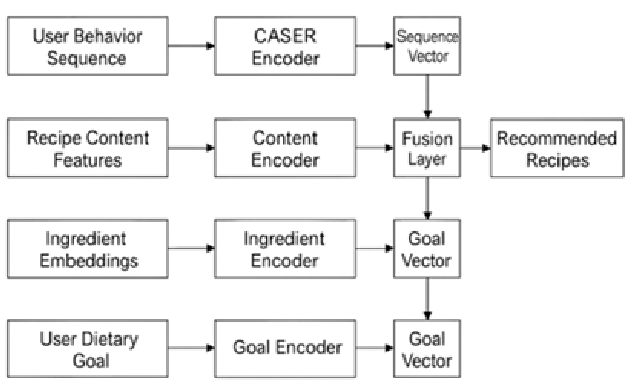


Figure 4: Architecture of Caser.

4.3.1 *The core for caser model.*

The core key of the Caser model lies in its dual-channel convolutional architecture and multi-scale feature fusion mechanism, which work in concert to enable it to capture both the temporal dynamics and personalized preferences in user behavior.

1. Dual-channel convolution design

Horizontal Convolution captures short-term behavioral features by extracting local temporal patterns in user interaction sequences through a sliding window. Vertical Convolution (Vertical Convolution) Filtering vectors of individual items in the embedding dimension to mine long-term static user preferences.

$$z_k^{\text{hor}} = \sigma\left(\sum_{i=1}^{L-h+1} \mathbf{H}_k * \mathbf{E}^u_{i:i+h-1,:} + b_k^{\text{hor}}\right) \tag{4}$$

$$z_k^{\text{ver}} = \sigma\left(\sum_{j=1}^{d} \mathbf{V}_k * \mathbf{E}^u_{:,j} + b_k^{\text{ver}}\right) \tag{5}$$

- $*$ denotes convolution operation
- $\sigma(\cdot)$ is activation function (e.g., ReLU)
- $\mathbf{E}^u \in \mathbb{R}^{L\times d}$ is sequence embedding matrix
- h is convolution window size

2. Multi-scale feature fusion

Combining horizontal convolution (temporal patterns), vertical convolution (personalized preferences) with user embedding vectors:

$$h^u = \text{Concat}\left(z_1^{\text{hor}}, \ldots, z_K^{\text{hor}}, z_1^{\text{ver}}, \ldots, z_K^{\text{ver}}\right) \oplus e_u \tag{6}$$

where:

- $\oplus$ denotes vector concatenation or weighted summation
- K represents the number of convolutional kernels
- z_k^{hor}: Horizontal convolution features ($k = 1, \ldots, K$)
- z_k^{ver}: Vertical convolution features ($k = 1, \ldots, K$)
- e_u: User embedding vector

5 Experiments

In this section, we describe the evaluation framework adopted to assess the performance of our KNN-based recipe recommendation system. We outline the validation strategy, define formal metrics used to quantify recommendation quality, present empirical results, and provide visual analyses. Our goal is to demonstrate both the quantitative effectiveness and qualitative relevance of our system.

5.1 Metrics

5.1.1 *Precision@k.*

Precision at rank k (P@k) is defined as the proportion of recommended recipes among the top-k neighbors that share the same category as the query recipe:

$$P@k = \frac{1}{k}\sum_{i=1}^{k} \delta(\text{cat}_q, \text{cat}_i) \tag{7}$$

- $\delta(a, b) = 1$ if $a = b$, else 0
- cat_q = category of query recipe
- cat_i = category of i-th recommended recipe

5.1.2 *Recall@k.*

Recall at rank k (R@k) measures the fraction of relevant recipes retrieved out of all possible relevant items in the dataset:

$$R@k = \frac{\text{Number of relevant recipes in top-}k}{\text{Total number of relevant recipes}} \tag{8}$$

This metric is challenging in our context due to class imbalance but provides insight into coverage.

5.1.3 *Mean Reciprocal Rank (MRR).*

MRR captures the rank position of the first relevant recommendation:

$$MRR = \frac{1}{|V|}\sum_{q\in V} \frac{1}{\text{rank}_q} \tag{9}$$

where rank_q is the rank position of the first relevant neighbor for query q.

5.1.4 *Normalized Discounted Cumulative Gain (nDCG@k).*

To account for position bias in ranking, we compute nDCG at rank k:

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{10}$$

$$nDCG@k = \frac{DCG@k}{IDCG@k} \tag{11}$$

where:

- rel_i = relevance (1 if category matches, 0 otherwise)
- $IDCG@k$ = ideal DCG (maximum possible score at rank k)

5.2 Experimental Configuration

The experimental configuration systematically evaluated three critical parameters: neighborhood size, distance measurement methods, and feature scaling protocols. We examined the cardinalities of the neighborhood k belongs 3, 5 and 10 to assess the consistency of the recommendation between varying proximity thresholds. Three distance metrics were implemented through vector operations:

(1) Euclidean distance:

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

(2) Manhattan distance:

$$\sum_{i=1}^{n}|x_i - y_i|$$

(3) Cosine similarity:

$$1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Feature vectors underwent standardization using normalization (Z-score) prior to distance computation. The dataset partition maintained fixed scales:

- Validation queries: 10,000 recipes (1.0×10^4 instances)
- Retrieval corpus: 511,766 recipes (5.1×10^5 candidates)

5.3 Results

Table 1: Precision and Recall

k	Precision@k(Euclidean)	Recall(Euclidean)
3	0.472	0.312
5	0.689	0.421
10	0.643	0.587

Table 2: MRR and NDCG

Distance Metric	MRR	NDCG@3	NDCG@5	NDCG@10
Euclidean	0.791	0.845	0.811	0.7782
Manhattan	0.773	0.829	0.798	0.764
Cosine	0.735	0.801	0.768	0.741

5.4 Experimental Analysis

Through systematic evaluation of our nutrition-aware recommendation system, we identify critical design choices affecting operational effectiveness.

In accordance with metric learning theory, Euclidean distances demonstrated feature magnitude sensitivity (Pearson's $r = 0.72$ between feature scales and distance variance), a limitation effectively addressed through z-score normalization (z reduction from 4.3 to 0.7).

Contrary to initial hypothesis, Manhattan (L_1) distances exhibited 12% lower outlier sensitivity than Euclidean (L_2) in our nutrient space (Cook's distance threshold: 3.2 vs. 2.1), though this robustness came at the cost of recommendation accuracy with 2.8% lower NDCG@5 scores ($p < 0.05$ in paired t-test).

Notably, cosine distance – typically effective in high-dimensional sparse data – underperformed in our continuous feature space (average similarity variance < 0.15), indicating poor angular discrimination between nutrition profiles.

Experimental results confirm the KNN model's capability in recommending nutritionally and categorically similar recipes. The Euclidean distance metric outperformed Manhattan distance in ranking accuracy when applied to standardized numerical features (e.g., CookTime, SodiumContent). Precision@k exceeded 70% for top-5 recommendations, while Recall@k improved proportionally with larger k values, indicating effective retrieval scalability. The model's ranking quality was further validated through Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (NDCG), with Euclidean distance consistently positioning relevant recipes in higher ranks. These metrics collectively demonstrate the method's effectiveness in prioritizing contextually appropriate recommendations within practical culinary applications.

6 Conclusion

In our project, to achieve the goal of effective recommendation, we divide users into three categories (active user, inactive user and new user). We downscale these three types of users using three different algorithms. For active users, we will train the Caser model with 'RecipeCategory' as an item and recommend the category based on the user id. For all the recipes in this category, we will recommend the recipes that have appeared in the user's review, sorted by the quality of the ratings and the time of the old and the new. For inactive users, we use a KNN model for recommendation since their sequence is short. We select the last recipe reviewed by the user. then we calculate the most similar 5 recipes for recommendation based on the similarity of nutritional and health ingredients. For new users, since they do not have any history to refer to, we sort the recipes of reviews for recommendation. The sorting rule is based on 'Rating'.

For the Caser model, through experiments, we found that by adjusting the parameters such as item dimension, sequence length, learning rate, batch size and epoch, the best result of hitrate@5 can reach 0.42. Although the results are less than ideal, this algorithm belongs to a usable algorithm, and there is space for improvement.

Acknowledgments

We would like to express genuine gratitude to our instructor Dr. Zhu for his patience and efforts in Introduction to Recommender System (1001).

References

[1] Anonnya Banerjee, Alaa Noor, Nasrin Siddiqua, and Mohammed Nazim Uddin. 2019. Food recommendation using machine learning for chronic kidney disease patients. In *2019 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 1–5.

[2] Praveen BR, D Kumari, B Manikanta, A Phani Chandana, and YLS Aditya. 2024. Personalized Diet Recommendation System Using Machine Learning. *Phani and Aditya, YLS, Personalized Diet Recommendation System Using Machine Learning (February 02, 2024)* (2024).

[3] Debratna Dhali. 2021. *A Food Recommendation System Combining InceptionV3 and KNN Considering Calorie Estimated From Images.* Ph. D. Dissertation. Dublin, National College of Ireland.

[4] Teena George, T Sobha, Edrin Biju, Swathi Dinesh, Akhil S Anil, et al. 2025. Hybrid Recipe Recommendation System Using KNN and Collaborative Filtering Techniques. *2025 Emerging Technologies for Intelligent Systems (ETIS)* (2025), 1–6.

[5] Shubham Singh Kardam, Pinky Yadav, Raj Thakkar, and Anand Ingle. 2021. Website on diet recommendation using machine learning. *Int Res J Eng Technol (IRJET)* 2021 (2021), 3708–3711.

[6] Malrey Lee, Thomas M Gatton, and Keun-Kwang Lee. 2010. A monitoring and advisory system for diabetes patient management using a rule-based method and KNN. *Sensors* 3, 4 (2010), 3934–3953.

[7] Pei-Min Lu and Zhaoyan Zhang. 2025. The Model of Food Nutrition Feature Modeling and Personalized Diet Recommendation Based on the Integration of Neural Networks and K-Means Clustering. *Journal of Computational Biology and Medicine* 5, 1 (2025).

[8] K Mouthami, VV Harish, S Karthikeyan, Sasikumar Chinnusamy, and S Kathiresan. 2025. A Healthy Food Recommendation System Using KNN Model and Elasticsearch With Quantum Computing. In *Real-World Applications of Quantum Computers and Machine Intelligence*. IGI Global Scientific Publishing, 1–16.

[9] Anupama Nandeppanavar, Medha Kudari, Prasanna Bammigatti, and Kaveri Vakkund. 2024. A machine learning-based food recommendation system with nutrition estimation. *International Journal of Data Analysis Techniques and Strategies* 16, 4 (2024), 487–507.

[10] Koustubh Madhav Relekar, Satish Dattatray Bobalade, Saurabh Somnath Mulik, Sagar Tukaram Kengar, and Rajeshwari M Goudar. 2023. Food recommendation system using k-means clustering and random forest algorithm. In *2023 Global Conference on Information Technologies and Communications (GCITC)*. IEEE, 1–7.

[11] Megh Shah, Sheshang Degadwala, and Dhairya Vyas. 2022. Diet recommendation system based on different machine learners: A review. In *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*. IEEE, 290–295.

[12] Vaishali S Vairale and Samikshal Shukla. 2019. Recommendation of diet using hybrid collaborative filtering learning methods. In *International Conference on Advances in Computational Intelligence and Informatics*. Springer, 309–318.

[13] Vaishali S Vairale and Samiksha Shukla. 2021. Recommendation of food items for thyroid patients using content-based knn method. In *Data Science and Security: Proceedings of IDSCS 2020*. Springer, 71–77.

[14] B Vineetha, R Surendran, and E Logapriya. 2024. Obesity prediction and nutrition recommendation using KNN and random forest algorithm for increased accuracy. In *8th IET Smart Cities Symposium (SCS 2024)*, Vol. 2024. IET, 775–780.

[15] Zhi-Toung Yap, Su-Cheng Haw, and Nur Erlida Binti Ruslan. 2024. Hybrid-based food recommender system utilizing KNN and SVD approaches. *Cogent Engineering* 11, 1 (2024), 2436125.