The object of this study is the process of creating a personalized menu. The subject of the study is recommendation systems for generating breakfast, lunch, snack, and dinner menus. The task solved was the development of an effective system for supporting the decisions by a wide range of users in planning a balanced diet. To form a menu of dishes of different categories of meals in a hybrid system for planning a balanced human diet, it is proposed to use different recommendation systems based on different models of artificial intelligence. The choice of the singular matrix decomposition model, the gradient boosting model of decision trees, and the wide and deep learning models for recommendation systems for forming a menu of dishes has been substantiated by the results of analysis. Based on the results of the experiment with these artificial intelligence models, it was determined which of them are more effective in solving the problem of forming a menu of meals for different categories of meals. The effectiveness of all models was evaluated by such test indicators as Precision@K, mean absolute and root mean square error. The feasibility of choosing the singular matrix decomposition model for generating breakfast menus and the wide and deep learning models for generating snack, lunch, and dinner menus was evaluated by the Precision@K values. The singular matrix decomposition model, compared to the other models studied in this paper, showed the highest Precision@K for breakfast, namely 0.942. The wide and deep learning models demonstrated the highest Precision@K for lunch, snack, and dinner: 0.961, 0.977, and 0.951, respectively. In practice, the results could be used to develop highly efficient personalized meal planning services in mobile and online platforms

Keywords: decision tree, deep learning, efficiency, matrix factorization, recommender system

UDC 004.8
DOI: 10.15587/1729-4061.2025.322316

IMPROVING OF INTELLIGENT DECISION SUPPORT SYSTEMS FOR PLANNING A BALANCED DIET

Viktor Ladyzhets
Corresponding author
PhD Student
Department of Information Technology Design and Applied Mathematics*
E-mail: victor.ladyzhets@gmail.com

Svitlana Terenchuk
Associate Professor
Department of Information Technology Design and Applied Mathematics*

Iryna Aznaurian
Associate Professor
Department of Physics*

Antonina Makhynia
Associate Professor
Department of Language Training and Communication*

*Kyiv National University of Construction and Architecture
Povitrianykh Syl ave., 31, Kyiv, Ukraine, 03037

Received 21.11.2024
Received in revised form 14.01.2025
Accepted 27.01.2025
Published 21.02.2025

How to Cite: *Ladyzhets, V., Terenchuk, S., Aznaurian, I., Makhinya, A. (2025). Improving of intelligent decision support systems for planning a balanced diet. Eastern-European Journal of Enterprise Technologies, 1 (3 (133)), 37–47. https://doi.org/10.15587/1729-4061.2025.322316*

1. Introduction

Statistical data on the growing percentage of overweight people worldwide indicate an urgent need to implement effective solutions for balanced nutrition [1–3]. Also, the results of a study of problems related to the diet of modern people have shown that an unbalanced diet and non-compliance with medical dietary recommendations significantly affect the quality and duration of life of a modern person [3].

A study conducted on data collected in the USA and Australia [4]:

– shows how overweight affects cardiovascular diseases;

– suggests that the increase in the percentage of overweight youth is already a social problem that may become much deeper in the coming years.

At the same time, there is a steady growing trend of young people's attraction to issues of caring for their health and independently finding solutions to regulate their personal diet [5].

At the same time, modern tools and applications for meal planning still do not provide personalized and adaptive recommendations that meet the personal needs and preferences of many people.

That is why the topic of developing intelligent systems remains relevant:

– supporting decisions on planning a balanced human diet;

– recommendations that could help plan a diet for a wide range of users who seek to independently regulate personal dietary restrictions, nutritional needs, and preferences.

2. Literature review and problem statement

In modern research, there are several approaches to building decision support systems that could facilitate menu selection and meal planning.

In [6], the CORA system is presented, which uses large language models (LLMs) to build a dialog system with a memory for recommending meals. CORA generates a list of ingredients for dishes that the user may like the most, based on the frequency of consumption of certain meals in the last few days, taking into account the best and worst ratings. The selection of a dish from the system database is carried out based on the user's answers to certain questions of the system, which provides feedback. Also in [6], the results of a study of the feasibility of using LLMs for recommending dishes are reported; it is shown that they can provide an acceptable level of user satisfaction with recommendations and ease of use.

However, in [6], issues related to the following remain unresolved:

– limited use of the user's personal data due to their reluctance to disclose detailed information about his diet;

– individual differences in users' communication styles.

These limitations and differences lead to the formation of incomplete profiles and insufficiently high-quality recommendations. An option to solve this problem may be to indirectly collect and analyze user data, without requesting this data in a dialog format.

In [7], such a solution is presented and investigated.

In the work, the authors used:

– variational autoencoder for transforming the user profile;

– recurrent neural network for generating recommendations;

– LLM ChatGPT for improving the generated recommendations, expanding possible dishes, and offering alternatives.

The results of study [7] showed that LLM could improve the accuracy of recommendation systems; however, the study also noted that LLMs are prone to errors and inaccuracies regarding the calorie content of food, portions, and the presence of allergens in the meal. These errors and inaccuracies are caused by the fact that currently no LLMs have been created that could significantly reduce the level of incorrect answers during generation and validate their results. An alternative option for overcoming these errors and inaccuracies of LLMs is the use of adaptive systems built on the basis of expert knowledge.

An approach based on expert knowledge and rules for different groups of users was used in [8]. This paper presents the PROTEIN AI Advisor system, which generates a meal plan taking into account both the user's dietary restrictions and preferences, as well as the rules defined by experts.

In this case, the following issues remain unresolved:

– lack of feedback from the user;

– inability to adapt recommendations to the dynamics of their preferences;

– poor scalability of the system.

The problem of adapting recommendation systems based on expert knowledge to the dynamics of user demand is solved by using artificial intelligence models that are able to take feedback into account.

This approach was used in [9] to develop the SHARE system, which combines:

– collaborative filtering to identify user preferences by analyzing the similarity between their preferences;

– content-based filtering to recommend recipes to the user that are similar to those they liked previously.

This combination allows SHARE to recommend menus and a ready-made weekly meal plan that match the user's personal tastes based on collective preferences.

Research conducted in [9] showed that SHARE is able to adapt to changes in user requirements using only the history of food choices collected indirectly from the user. However, scalability and efficiency problems remain, which are associated with the need to manipulate large data sets, which is complicated by the use of the K-nearest neighbors (KNN) method in SHARE.

Using KNN:

– leads to slow recommendation generation;

– makes it impossible to take into account such hidden factors as subtle taste preferences or dietary needs, which are the factors that underlie user preferences;

– makes it impossible to take into account such contextual and related information as user demographics and dish attributes.

Studies [10, 11] solve the problem of working with large data sets. In these works, a modified version of the KKN model and the Gated Recurrent Unit (GRU) model were tested for recommendation systems. The test results showed that both models require further research to improve the personalization of recommendations, efficiency, and adaptability.

In [12] it is shown that deep learning models are relevant for the task of building recommendation systems due to their power in processing large amounts of heterogeneous information. Deep neural networks can learn high-level abstractions, which provides accurate and personalized recommendations. This area is actively researched and developed, as a result of which new architectures and methods appear that improve the performance of recommendation systems.

In work [13], which focuses on improving the structure of nutrition recommendation systems:

– a three-level architecture of a recommendation system for menu generation is proposed;

– the Wide&Deep model is tested, which combines the advantages of linear models (wide part) and deep neural networks (deep part).

This combination allows Wide&Deep to simultaneously take into account both general patterns and complex relationships in the data. Compared to traditional deep neural networks and autoencoders, this model is more efficient when working with sparse data and requires fewer computational resources, which simplifies its development and implementation. The results of study [13] showed the feasibility of using the Wide&Deep deep learning model for nutrition recommendation systems, but a significant improvement in the efficiency of such systems still remains an open problem.

Thus, despite their innovation, existing decision support systems for diet planning are not yet able to provide highly personalized and adaptive planning of balanced nutrition. Also, these systems are not yet able to take into account the complex dietary restrictions and needs of users at an acceptable level.

The main objective reason for such limitations is the reliance of traditional approaches to diet planning on static data that do not take into account the dynamics of life, preferences, and health status of users. One of the main subjective reasons for the difficulty of creating and implementing a universal recommendation system into people's everyday lives is dissatisfaction with the system when personal requirements deviate from the normal distribution. Such deviations for one or another menu selection criterion are explained by training models on general data sets. Because of this generalization, all systems are not sufficiently personalized to take into account the unique requirements and preferences of the user.

The difference between personalized recommendation systems and prediction systems is that the former return recommendations in descending order of the probability of the user liking the recommended item.

To assess the effectiveness of recommendation systems, Precision@K or Recall@K are usually used, where:

– Precision@K shows what percentage of the first K recommendations is relevant, i.e., corresponds to the user's interests [14];

– Recall@K shows what part of the entire possible relevant set the system was able to find among these recommendations [15].

Since users tend to pay attention to only a few of the best suggestions, it is more important to ensure that these suggestions are of high quality. Thus, Precision@K simulates real user behavior, which usually considers only the very first recommendations [14]. This approach helps improve the user experience and has a positive impact on business indicators since the first recommendations are key to customer acquisition and conversion. Moreover, under real-world conditions, relevance data is often incomplete, which makes Precision@K a more practical metric. That is why in this work, Precision@K is chosen as the main criterion for evaluating the effectiveness of menu recommendation systems.

3. The aim and objectives of the study

The aim of our research is to improve intelligent decision support systems for personalized planning of a balanced human diet in order to improve the efficiency of the system for recommending menus for different categories of meals using the Precision@K metric. This will significantly increase the efficiency of decision support systems as a whole.

To achieve the goal, the following tasks need to be solved:

– to propose the architecture of a personalized intelligent system for planning a balanced diet;

– to conduct an experimental study of candidate models for recommendation systems for all categories of meals;

– to justify the choice of models for recommendation systems for breakfasts, lunches, snacks, and dinners using the Precision@K criterion.

4. The study materials and methods

4. 1. The study materials

The object of our study is the process of forming a personalized menu by an intelligent system for planning a balanced diet. The subject of the study is recommender systems used to generate breakfast, lunch, snack, and dinner menus.

To train the personalized menu recommendation systems, the Food.com input data set [10] was used, which was provided by the Food.Com recipe web service, and contains more than:

– 500,000 recipes from users;

– 1,400,000 user ratings of recipes.

The choice of Food.com is justified by its accessibility to a wide range of users and its suitability for practical and experimental research according to the requirements described in [16]. The main drawback of Food.com, which leads to problems, one of which is the impossibility of evaluating the recommendation system on a large number of predictions due to the limited number of recipe ratings available to each user, is data sparsity. Therefore, the menu division into breakfasts, lunches, dinners, and snacks was carried out from the 100,000 most popular data sets. It was assumed that one recipe could belong to several categories at the same time (Fig. 1), and a certain percentage of recipes that did not belong to any of the categories were defined as others.

From the 100,000 most popular datasets, we obtained:

– 7,133 recipes that were labeled as breakfast;

– 15,872 recipes that were labeled as lunch;

– 15,579 recipes that were labeled as snacks;

– 42,165 recipes that were labeled as dinner.

However, despite the fact that these categories overlap somewhat, a significant number of recipes belong to only one category, which indicates that each type of food is significantly different from the others (Fig. 1).

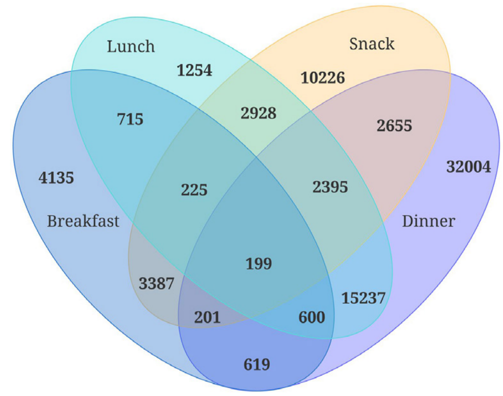

Fig. 1. Venn diagram of the distribution of the dataset by meal categories

The distribution of recipes by meals allows us to form subsets of data for separate training of recommendation systems for each meal.

For each recipe, a set of tags was defined that indicate the recipe's belonging to a certain type of food. Currently, the following types have been considered: {vegan, vegetarian, sugar-free, lactose-free, gluten-free, keto, paleo, low-fat, low-carb, high-protein}.

Also, for each recipe, the allergens present in the list were determined: {gluten, peanuts, tree nuts, dairy, eggs, soy, wheat, fish, shellfish}. In addition, the Food.com dataset contains other data about the recipes that were useful in building the recommendation system: {preparation time, calorie content and technique required to prepare the dish}.

The dataset used in this study contains recipes and user ratings collected over 18 years.

The data was randomly divided into a training (70 %) and a test (30 %) sample. Only recipes with at least 20 user reviews were selected for both samples. Only users who rated at least 20 recipes from the training and test samples were selected. This ensures that each user has the opportunity to test the accuracy of the system on the test sample. All records in the training and test samples are complete and checked for inconsistencies.

The user feedback dataset record, which is the input vector of the experimental study, has the form: {RecipeId, UserId, Rating, Techniques, IsBreakfast, IsLunch, IsSnack, IsDinner}, where:

– RecipeId and UserId – identifiers of the recipe and the user who left the review, respectively;

– Rating – user rating;

– IsBreakfast, IsLunch, IsSnack and IsDinner – Boolean values indicating whether the recipe belongs to the corresponding meal;

– Techniques – JSON string containing an array of Boolean values indicating whether the corresponding cooking technique is used in this recipe.

The set of cooking techniques that were considered in the construction of the recommendation systems as recipe characteristics consists of: {barbecuing, blanching, boiling, leveling, holding, pot cooking, glazing, deactivating, distilling, emulsifying, freezing, kneading, salting, baking, thickening, pouring, mixing, caramelizing, fermenting, combining, smoking, marinating, dicing, chilling, scalding, slow cooking, pureeing, raising, microwaving, pressure cooking, straining, grinding, straining, grilling, toasting, mashing, smoothing, rubbing, softening, mashing, melting, pickling, pan frying, grilling, stewing, braising, steaming, fermenting}.

The user reviews datasets for breakfast, lunch, snack, and dinner are subsets of the overall user reviews dataset.

The user reviews dataset has 325,568 records, for 49,251 unique recipes, and 99,338 unique users, with:

– breakfast set contains 47,981;

– lunch set contains 90,398;

– snack set contains 86,762;

– dinner set contains 120,427 records.

The dataset presents all ratings on a scale from 1 to 5. It was taken into account that different users have different understandings of the rating scale and a tendency to overestimate or underestimate ratings, and to minimize the influence of subjectivity, normalization of user ratings of dishes was carried out.

The main hypothesis of the study on improving the efficiency of balanced diet planning systems is based on the assumption of improving the efficiency of menu recommendation systems for each type of meal when they are trained separately.

Separate training of menu recommendation systems should provide the balanced diet planning system with such advantages as:

– optimization of results according to various factors, since taste characteristics and techniques used during cooking have different weights for different categories of dishes;

– the ability to understand which models of nutrition recommendation systems show the best results;

– increasing the accuracy of recommendations due to the fact that the models better capture hidden connections for each meal;

– reducing the level of uncertainty and improving adaptation to the user's needs and preferences by distributing reviews by dish categories.

4. 2. Research environment and software

All models of menu recommendation systems were trained and tested in the Google Colab cloud environment, which allows for the prompt use of hardware acceleration without additional costs for deploying your own computing servers.

To develop the recommendation system, the surprise [16, 17], LightGBM [18], and tensorflow [19] libraries were used, as well as such input data as user preferences, product features, and contextual information.

During the experiments:

– the main programming language was Python, version 3.11;

– an NVIDIAT4 GPU with 16 GBGDDR6 RAM was used, which provides a sufficient supply of computing power even for deep learning models;

– pandas and numpy libraries were used to work with data;

– entire program code was structured in JupiterNotebooks.

Among the training methods, machine learning methods were selected, which are described in scientific publications on building recommender systems [20–22].

4. 3. Research methodology

The methodology of this experimental study of menu recommendation systems is based on four consecutive stages.

At the first stage, data selection and preparation are carried out, which involves:

– determining the source and volume of the sample;

– analyzing its quality and removing anomalous or incomplete records;

– filtering recipes with too few ratings.

In the second stage, data is divided into training, variation, and test subsamples, which provide an objective test of the model's ability to generalize.

In the third stage, models are trained on the training subsample using appropriate objective functions and metrics, which may include minimizing criteria based on the mean absolute deviation (MAE) or root mean square deviation (RMSE).

In the fourth stage, trained models are compared with data from the test subsample, which was not used during training. MAE and RMSE were used to assess the reliability of the prediction, but these metrics do not reflect the ranking and selective nature of user interaction with the recommendation lists. In this case, users of recommendation systems usually study a short list of recommended meals and their experience is shaped by the presence or absence of relevant content in this limited set.

Precision@K provides an assessment of the efficiency of recommendation systems by the proportion of relevant elements in the first K recommendations presented to the user by the system [14]:

$$P = NR/K, \tag{1}$$

where NR is the number of relevant recommendations (rr); K is the number of best recommendations (br).

Items whose scores in the test subset exceed a given threshold are considered relevant, and Precision@K focuses on how many of the K best recommendations are truly relevant, thus reflecting the system's ability to provide accurate and useful advice.

Thus, Precision@K offers a more direct measure of the practical utility of a personalized recommendation system compared to MAE and RMSE. By focusing on the proportion of correctly found relevant items in the K top recommendations, Precision@K is more closely aligned with real-world decision-making processes, thereby providing a clearer indicator of the system's effectiveness in meeting user needs and expectations.

The simplification adopted in the work is that the system is oriented towards predicting the user's personal demand. At the same time, differences in preferences and constraints across user groups are ignored.

4. 4. Research models and methods

Various machine-learning methods were used in this work.

The matrix factorization method, on which collaborative filtering is based, involves representing the user rating matrix in the following way:

$$R \approx P \times Q^T, \tag{2}$$

where R is the user rating matrix of size $U \times I$; P is the user-factor matrix of size $U \times d$; U is the number of users; Q is the element-factor matrix of size $I \times d$; I is the number of recipes; d is the number of latent factors.

For the u-th user and the i-th dish, the predicted rating is determined by the formula:

$$\hat{r}(u,i) = p_u \cdot q_i, \tag{3}$$

where $\hat{r}(u,i)$ is the predicted rating for the u-th user and the i-th recipe; p_u is the u-th row of the matrix P, q_i is the i-th row of the matrix Q in (2).

To find P and Q, the stochastic gradient descent method is used, and the parameters p_u and q_i are updated iteratively in the direction of the negative gradient of the loss function for each observed rating.

The MSE loss function with L2-regularization to avoid overfitting has the form:

$$Loss = \min_{P,Q} \sum\nolimits_{(u,i)\in\Omega} \left(r_{u,i} - p_u \cdot q_i\right)^2 + \\ + \lambda\left(\sum\nolimits_u |p_u|^2 + \sum\nolimits_i |q_i|^2\right), \qquad (4)$$

where Ω is the set of known user-recipe pairs; $r_{u,i}$ is the actual score for the u-th user and the i-th recipe; λ is the regularization parameter; $|p_u|$ is the Euclidean norm (L2) of the latent vector of the user p_u.

The SVD model is trained for a given number of epochs, which is one of the key hyperparameters of the model. The SVD model can stop training early if the RMSE metric does not improve over several epochs. In this work, during the experimental study, the number of epochs for all trained SVD models was 30, and the number of latent factors d was 100.

Gradient Boosted Decision Trees (GBDT) is a model of recommendation systems that uses an ensemble method in which several decision trees are sequentially trained, and each subsequent ensemble of trees must correct the errors of the previous one. Each tree is a sequence of feature splitting that split the data in a way that best predicts the residuals of the current iteration. The goal of training is to predict the user's clear evaluation of the recipe.

The loss function is defined as:

$$Loss = \frac{1}{2}\sum\nolimits_{i=1}^{N} \left(y_i - F\left(x_i\right)\right)^2, \qquad (5)$$

where N is the number of elements in the training set (x_i, y_i); x_i is the user characteristics; y_i is the user's actual rating of the i-th recipe, including cooking techniques; $F(x_i)$ is the current GBDT forecast.

The GBDT forecast at the first iteration $F_0(x)$ is defined as the average of the actual ratings of the i-th recipe:

$$F_0\left(x\right) = \frac{1}{2}\sum\nolimits_{i=1}^{N} y_i. \qquad (6)$$

The GBDT forecast at the current (m–1-th) iteration is constructed as follows:

$$F_m(x) = F_{m-1}(x) + \gamma h_m(x), \qquad (7)$$

where F_m is the GBDT forecast at the current iteration; γ is the shrinkage parameter, taking values from 0 to 1; $h_m(x)$ is the m-th regression tree trained specifically to approximate the residuals of the current iteration.

The selected tree is added to the general model, approximating the forecasts to the true estimates.

The residuals of the current (m–1-th) iteration are calculated by the formula:

$$r_i^{(m)} = y_i - F_{m-1}\left(x_i\right), \qquad (8)$$

where $r_i^{(m)}$ – residuals of the current iteration; F_{m-1} – GBDT predictions at the current iteration; y_i – actual user score.

After M iterations, the final GBDT model for predicting scores has the form:

$$F_M\left(x\right) = F_0\left(x\right) + \sum\nolimits_{m=1}^{M} \gamma h_m\left(x\right). \qquad (9)$$

One of the key hyperparameters of the GBDT model is the number of epochs (M) during which it is trained. In this work, during the experimental study, M=200 was used for all GBDT models, but the model can stop training earlier if the RMSE metric does not improve over several epochs. The shrinkage parameter γ=0.05.

Wide&Deep is a hybrid model that combines:

1. A wide component to remember the recurrence and interaction of features.

2. A deep component to generalize and capture more complex relationships through learned embeddings and hidden layers.

The wide component is a linear model that captures explicit coincidences and interaction of features, often using cross-transformations or one-to-one encodings. This linearity helps the system remember frequently occurring patterns, such as a user's preference for a particular food category.

The structure of the wide component of the Wide&Deep model, which was chosen for the experimental study, is shown in Fig. 2.

The input vector of the wide component consists of two main inputs: UserId and RecipeId, each of which is mapped to an embedding that transforms these categorical variables into dense one-dimensional vectors. These embeddings are then multiplied element-wise, capturing the interaction between the features. The resulting vector is smoothed into a single vector that is passed through a linear layer. This layer computes a weighted sum of the inputs, outputting a scalar value. The simplicity of this architecture allows the Wide component to effectively model direct associations between user and recipe characteristics.

The deep component is a multilayer neural network that learns dense embeddings and captures complex, nonlinear relationships. This component generalizes hidden or rare combinations of user and dish characteristics, mapping them into latent representations.

When training the Wide&Deep model, ensuring the reliability of rating prediction involves minimizing the discrepancy between the predicted and actual ratings using MSE(4) with L2 regularization, which is used to avoid overtraining.

The structure of the deep component of the Wide&Deep model, which was selected for experimental research, is shown in Fig. 3.

The input vector of the Wide&Deep deep component consists of: UserId, RecipeId and Techniques, which represent unique identifiers of authors and recipes, as well as a vector of characteristics of the techniques used to prepare the dishes.

For categorical inputs UserId and RecipeId, this Wide&Deep component uses separate embedding layers that transform these sparse identifiers into dense 32-dimensional representations.

After each embedding layer, a smoothing operation is performed that transforms the two-dimensional embedding results into one-dimensional vectors.

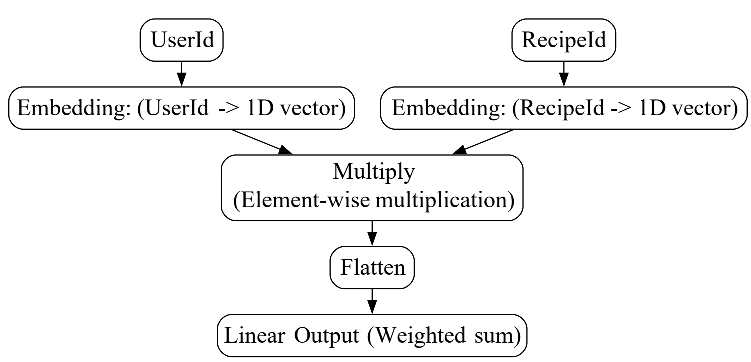

Fig. 2. Structure of the wide component of the Wide&Deep model

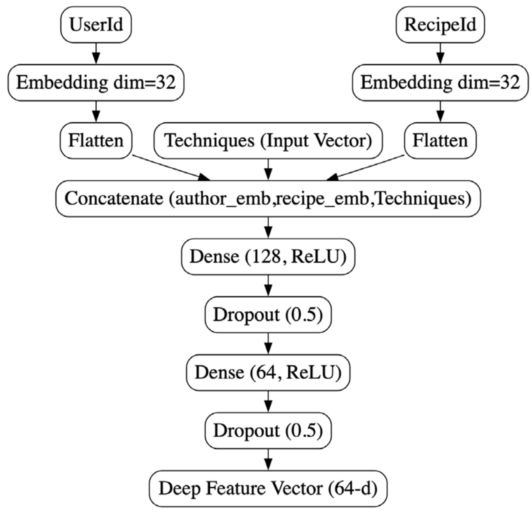

Fig 3. Structure of the deep component of the Wide&Deep model

The raw recipe techniques input data are not embedded but are used directly as categorical data.

The smoothed embeddings and the techniques vector are combined using concatenation, forming a single feature vector that contains all the necessary information.

This combined feature vector is fed to a series of hidden layers, where:

– it passes through a dense layer with 128 neurons using the ReLU activation function, which captures nonlinear relationships between features;

– it passes through a dropout layer with a dropout factor of 0.5 to reduce overfitting by randomly deactivating half of the neurons during training;

– the output of the first dropout layer is processed by a second dense layer with 64 neurons, also using ReLU activation, followed by another dropout layer with a dropout factor of 0.5;

– the final result of the deep component is a 64-dimensional feature vector that represents the learned high-order feature interactions. These features contribute to the overall prediction when combined with the wide component.

Wide&Deep is trained for M epochs, which is one of the hyperparameters of the model. In this work, during the experimental study, $M=20$, but the model may stop training earlier if the RMSE metric does not improve over several iterations.

5. Results of research on intelligent decision support systems for planning a balanced diet

5. 1. Architecture of an intelligent system for personalized planning of a balanced diet

Fig. 4 shows the architecture of an intelligent system for personalized planning of a balanced diet, which allows the use of separately trained menu recommendation systems for each meal: breakfast, lunch, snack, and dinner.

When developing the architecture of this system, the main attention was paid to the consistent formation and adaptation of the user's individual nutrition plan based on:

– analysis of their personal data;

– specific restrictions and goals regarding body weight;

– division of the recommendation process into several specialized components for each type of meal.

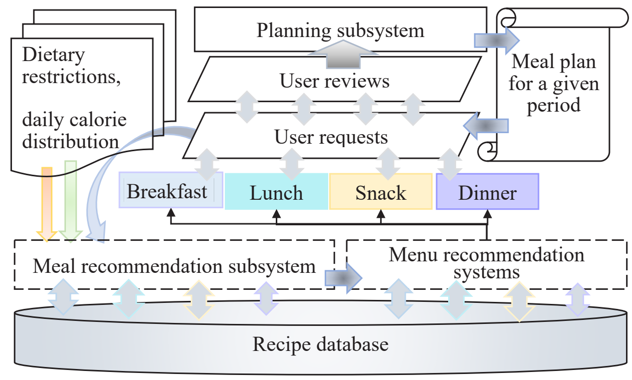

Fig. 4. Architecture of an intelligent system for personalized planning of a balanced diet

Therefore, the system under development begins its work by receiving initial data from the user regarding:

– goal (weight loss, maintenance, or weight gain);

– dietary restrictions (vegetarian, gluten-free, lactose-free, etc.);

– daily calorie intake.

Based on this data, the meal recommendation subsystem determines the approximate distribution of calories for breakfast, lunch, snack, and dinner.

In this work, the following distribution is proposed by default:

– breakfast: 25 % of the daily calorie intake;

– lunch: 35 % of the daily calorie intake;

– snack: 10 % of the daily calorie intake;

– dinner: 30 % of the daily calorie intake.

According to this distribution, the menu recommendation system forms a selection of dishes from different categories of meals.

It is assumed that to form specific menu suggestions, the system uses several autonomous, separately trained recommendation models, each of which specializes in one of the categories of meals. This approach is motivated by the need to optimize recommendations for such different factors as taste characteristics, cooking techniques and the desire to increase the accuracy of predictions regarding user preferences.

At the same time, one of the requirements for recommendation systems is the ability to detect general patterns and complex relationships in the data since this makes it possible to take into account hidden factors underlying user preferences and thus:

– prevent dissatisfaction with the system when the user's personal preferences deviate from the normal distribution of preferences of other users;

– adapt recommendations to the dynamics of user preferences, minimizing their communication with the system.

Next, the planning subsystem, using the feedback received from the user, forms a meal plan for a given period. At this stage, the system analyzes user feedback on previously suggested dishes and their recipes. Taking into account this data makes it possible to reduce the level of uncertainty and increase the personalization of recommendations, as the diet plan is adjusted with changes in the preferences and possible additional restrictions of the user.

The result of the system is an integrated diet plan with recipes for dishes that take into account calorie content and the ratio of proteins, fats, and carbohydrates. After receiving the user's assessment and feedback, the system adapts future offers, taking into account the dynamics of their preferences or the identification of additional restrictions. Thus, the entire architecture of the system is aimed at the flexible formation of a personalized diet for the user.

5. 2. Experimental study of candidate models for recommendation systems for each type of meal

For the experimental study of candidate models for recommendation systems for each type of meal, the SVD, GBDT, and Wide&Deep models were selected because they:

– can detect complex relationships between the characteristics of meals and user preferences, which makes it possible to form recommendations even if individual tastes deviate from the mathematical expectations of the parameters found in the general data set;

– can work effectively with such complex and sparse information about meals as the use of different ingredients and cooking methods, taking into account the significant features of the data structure;

– cover various machine learning methods: matrix factorization, tree-like methods, and deep learning.

In addition, the presence of popular frameworks simplifies their implementation, configuration, and troubleshooting. Finally, the computational resources of these models allow for relatively flexible tuning of hyperparameters during experiments, without exceeding the available hardware limitations.

The model trained on a dataset containing all recipes and user ratings is called the universal configuration in this work.

The model trained on a dataset containing only breakfast, lunch, snack, or dinner recipes is called the specialized configuration in this work.

The dataset containing all recipes and ratings from [10] was used to train the universal configuration, and the datasets containing only recipes for one meal category were used to train the specialized configuration.

To test the hypothesis of improving the efficiency of recommendation systems during separate training, a study was conducted, during which:

– each of the SVD, GBDT, and Wide&Deep models was trained on 5 datasets;

– for breakfast, lunch, snack, and dinner dishes, personal recommendation tables were generated by all SVD, GBDT, and Wide&Deep configurations.

Each dataset was divided into training (70 % of the data) and test (30 % of the data). All models were trained on training models, and then their effectiveness was tested on test models.

To understand how relevant the first 10 recommendations are on average:

– after training all SVD, GBDT, and Wide&Deep configurations, the Precision@10 of each model was estimated for all users;

– based on these estimates, it was proposed to recommend the best system for each meal category.

Fig. 5 shows the dynamics of the MSE metric when training the universal and specialized SVD model configurations for breakfasts, lunches, snacks, and dinners.

The dynamics of learning different SVD configurations show that:

– models of all configurations for breakfast, lunch, and snack finished training on the 15th epoch;

– models of both configurations for dinner finished training on the 20th epoch.

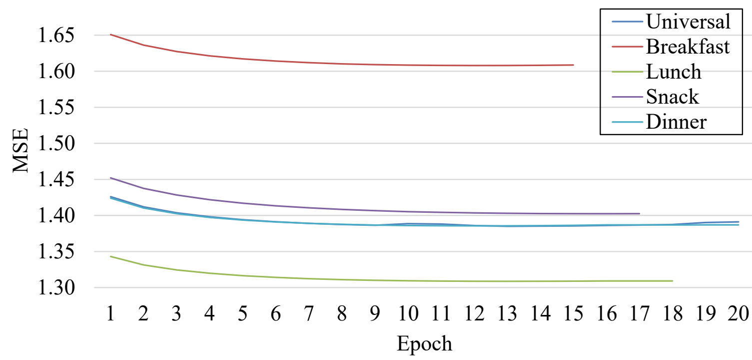

Fig. 5. MSE when training universal and specialized configurations of the SVD model

The example of the SVD model also shows the first 10 personal recommendations and ratings of breakfast meals provided to the user with id = 169430 by universal and specialized models of different configurations (Table 1).

Table 1

Example of breakfast meals provided to user with id = 169430, universal and specialized SVD configurations

SVD Configuration	No.	Meal ID	Score on a 5-point scale	
			Real	Predicted
Universal	1	Banana nut bread	5.0	5.0
	2	Creamy scrambled eggs	5.0	5.0
	3	Raisin spice hot cereal with quinoa	5.0	5.0
	4	Onion bread for the abm	5.0	4.9787
	5	Nutty banana oatmeal	5.0	4.9404
	6	Low-fat banana buttermilk muffins	5.0	4.9281
	7	Doctored up scrambled eggs	5.0	4.8805
	8	Country cornbread	5.0	4.7166
	9	Chocolate fudge power oatmeal	3.0	4.5393
	10	Cheese bacon breakfast muffins	3.0	4.3954
Specialized	1	Creamy scrambled eggs	5.0	5.0
	2	Low-fat banana buttermilk muffins	5.0	5.0
	3	Bacon egg and cheese biscuit	5.0	4.9888
	4	Baked Sicilian frittata	5.0	4.9470
	5	Tabasco cheddar biscuits	5.0	4.8864
	6	Broccoli cheese frittata	5.0	4.8736
	7	Orange almond scones	5.0	4.8683
	8	Banana nut bread	5.0	4.8253
	9	Scrambled egg wrap	3.0	4.3265
	10	Country cornbread	5.0	4.2023

The data in Table 1 show that at a given relevance threshold of 4.0:

– the universal model recommended 8 out of 10 relevant meals since the real rating is less than 4.0 for meals numbered 9 and 10;

– the specialized model recommended 9 out of 10 relevant meals since only dish numbered 9 has a real rating of less than 4.0.

Thus, the Precision@10 of the universal and specialized models for breakfast meals offered to the user with id = 169430 is 0.8 and 0.9, respectively. Similarly, tables for lunches, snacks, or dinners were formed, provided by the universal and specialized SVD models.

The average Precision@10 values of the universal and specialized SVDs trained to recommend meals of different food categories are given in Table 2.

Table 2

Comparison of average Precision@10 for different SVD configurations

Food intake category	Average Precision@10 of SVD model (rr/br)		
	Universal configuration	Specialized configuration	Improvement, %
Breakfast	0.7951	0.9423	18
Dinner	0.8058	0.9046	12
A snack	0.8069	0.8939	10
Dinner	0.7652	0.8856	15

The same experimental studies for all categories of meals were conducted for all GBDT and Wide&Deep configurations. Moreover, the process of building recommendations by these models includes various cooking techniques and assessing their impact on preferences.

Fig. 6 shows the dynamics of the RMSE metric at each epoch of training of the universal and specialized configurations of the GBDT model for breakfasts, lunches, snacks, and dinners.

The learning dynamics of different GBDT configurations show that:

– the model with the breakfast configuration showed the highest RMSE error among all configurations, including the universal one, while the lunch model showed the lowest error;

– the specialized models learn much faster, finishing training prematurely around 50 epochs, while the universal model finished training around 130 epochs, as the RMSE metric stopped improving.

The average Precision@10 values of universal and specialized GBDTs trained to recommend meals from different food categories are given in Table 3.

Fig. 7 shows the dynamics of the RMSE metric when training the universal and specialized configurations of the Wide&Deep model for breakfasts, lunches, snacks, and dinners.

The learning dynamics of different Wide&Deep configurations show that:

– all configurations finished training during the 11th epoch;

– the breakfast configuration showed the highest RMSE error, while the dinner configuration and the universal configuration showed the lowest values of this metric.

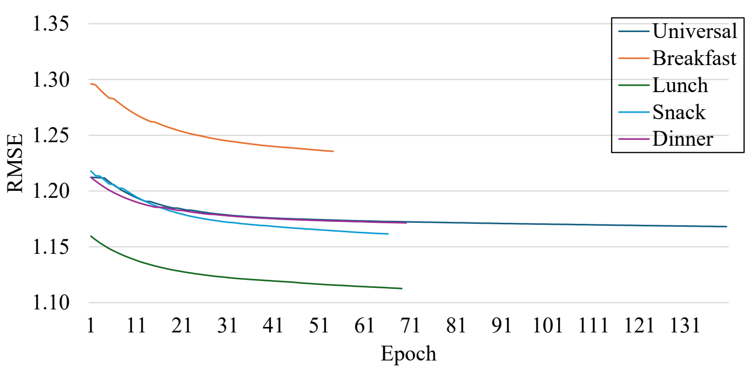

Fig. 6. RMSE when training universal and specialized configurations of the GBDT model

Table 3

Comparison of average Precision@10 for different GBDT configurations

Food intake category	Average Precision@10 of GBDT model (rr/br)		
	Universal configuration	Specialized configuration	Improvement, %
Breakfast	0.7115	0.8613	21
Dinner	0.6592	0.8331	26
A snack	0.7257	0.8479	17
Dinner	0.6372	0.7528	18

The average Precision@10 values of universal and specialized Wide&Deep, trained to recommend meals from different food categories, are given in Table 4.

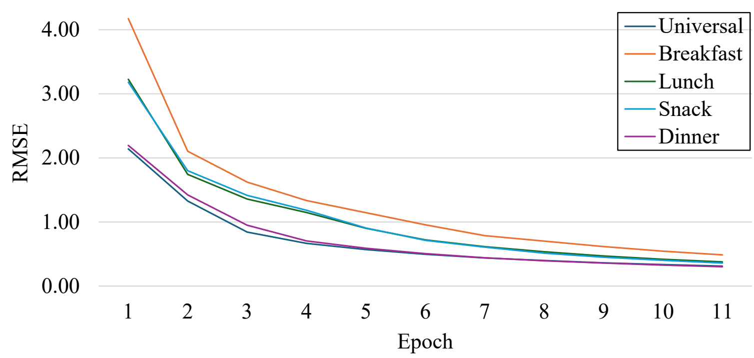

Fig. 7. RMSE when training universal and specialized configurations of the Wide&Deep model

Table 4

Comparison of average Precision@10 for different Wide&Deep configurations

Food intake category	Average Precision@10 of Wide&Deep model (rr/br)		
	Universal configuration	Specialized configuration	Improvement, %
Breakfast	0.7631	0.9210	20
Dinner	0.8155	0.9609	18
A snack	0.8154	0.9768	19
Dinner	0.7719	0.9509	23

Thus, as a result of separate training for each meal, the efficiency of work according to the Precision@10 metric improved for all models.

5. 3. Justification of the choice of models for breakfast, lunch, snack, and dinner menu recommendation systems according to the Precision@K criterion

Fig. 8 shows bar charts that demonstrate the average value of Precision@10 for the SVD, GBDT, and Wide&Deep models on the ordinate axis.

These diagrams show that according to the Precision@10 metric, GBDT demonstrates the worst results: only about 86 % of the first 10 breakfast meals recommended to the user are relevant, while the SVD and Wide&Deep models improve this metric for breakfast to 94 % and 92 %, respectively. The Wide&Deep model showed the highest efficiency compared to Precision@K for lunches, snacks, and dinners, recommending on average about 96 % of relevant meals for lunches, 97 % for snacks, and 95 % for dinners.

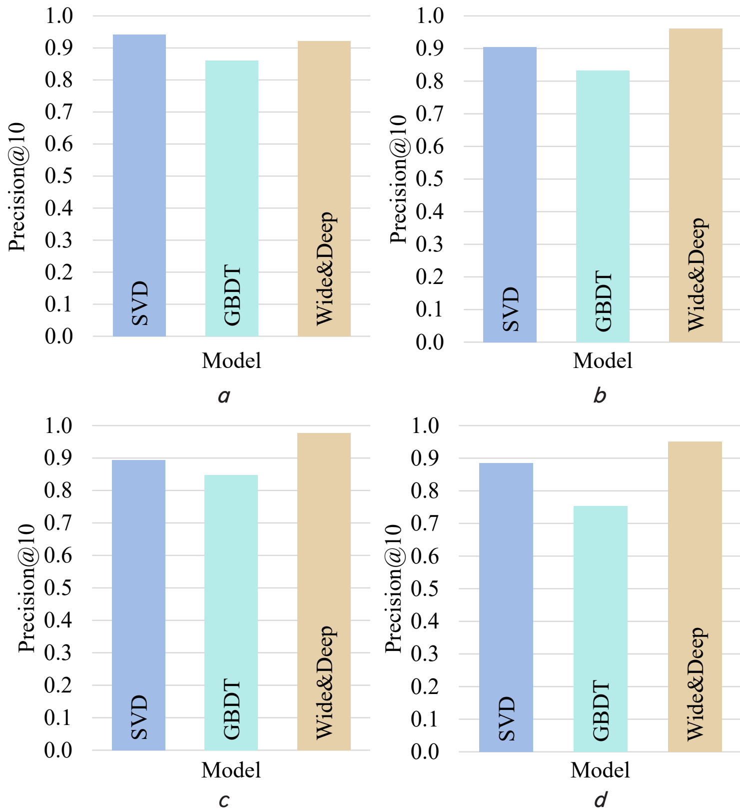

Fig. 8. Average Precision@10 charts of specialized SVD, GBDT, and Wide&Deep configurations for: *a* — breakfasts; *b* — lunches; *c* — snacks; *d* — dinners

Thus, the results of the experimental study provide grounds to argue that the feasibility of choosing the SVD and Wide&Deep models for further improvement of the intelligent system for personalized planning of a balanced diet has been sufficiently justified.

6. Discussion of results based on the experimental study of artificial intelligence models for personalized menu recommendation systems

One of the main scientific results of our work is the architecture (Fig. 4) of an intelligent system for personalized balanced diet planning, which, unlike:

– the CORA system [6], which forms a list of ingredients for meals that the user may like the most, offers a menu of the most relevant meals without requiring excessive communication;

– systems using LLM [7], which takes into account the calorie content of food, portions, and the presence of allergens when planning a personalized menu;

– the PROTEIN AI Advisor system [8], which forms a meal plan taking into account both the user's dietary restrictions and preferences, as well as the rules defined by experts, and adapts to dynamic changes in nutritional needs, analyzing their choices and feedback (Table 1);

– the SHARE system [9], can solve the scalability and efficiency problems associated with the need to manipulate large data sets.

This advantage of the personalized balanced diet planning system is ensured by the effective operation of autonomous menu recommendation systems, which:

– better adapt by using separate recommendation systems for different meals;

– better match user behavior than the deep learning model [12] and the Wide&Deep model [13], due to the use of the Precision@K metric to assess the effectiveness of the recommendation systems during training.

The results of the separate training process of candidate models for recommendation systems for each type of meal showed (Fig. 5–7) that:

– specialized SVD and GBDT models are trained for fewer epochs than the universal system and show no worse final metric values after training for all meals, except breakfasts and snacks for SVD and breakfasts for GBDT;

– universal and specialized Wide&Deep models were trained for the same number of epochs and showed similar final metric values.

The results of the experimental study of candidate models for recommendation systems for each type of meal showed (Tables 2–4) that separately trained specialized systems for breakfast, lunch, snacks, and dinner show an increase in the efficiency of work according to the Precision@10 metric compared to the universal system. The improvements ranged from 10 % (Table 2) to 26 % (Table 3). This increase is explained by the fact that specialized models are trained on more homogeneous sets of examples, thanks to which they are better able to detect patterns specific to the corresponding type of dish. Limited context means less conflicting or redundant information, which facilitates the tasks of searching and ranking recipes by relevance.

At the same time, a universal model, covering all types of meals, can benefit from a larger amount of data but loses some accuracy due to heterogeneity arising from mixing different categories of recipes.

Based on the results of comparing the effectiveness of SVD, GBDT, and Wide&Deep models by the Precision@10 metric for recommending breakfasts, lunches, snacks, and dinners (Fig. 3), the feasibility of using:

– SVD configuration for breakfast recommendation systems;

– Wide&Deep configuration for lunch, snack, and dinner recommendation systems has been justified.

The better performance of the SVD-based breakfast recommendation systems based on the Precision@10 metric is explained by the fact that the methods of preparing meals available in the dataset [10] have less influence on user preferences. The fact that the Wide&Deep-based system showed better results for other categories is explained by the fact that the combination of generalization and memorization has a positive effect on the performance of recommendation systems.

Among the limitations of our study, the following should be highlighted:

– insufficient number of recipe ratings by an individual user in the dataset [10], which can lead to errors when testing trained models;

– limited resources, which significantly narrows both the set of models for research and the number of parameter combinations on which the models were trained during the study.

However, at this stage of the study, due to the limited resources, menu recommendation systems have a number of shortcomings, namely, they do not take into account:

– eating out during planning;

– prices of recommended meals;

– demographic and cultural characteristics of users;

– preferences of the user group;

– seasonality of user preferences and popularity of holiday meals;

– trendiness of meals and recipes during the period of popularity they acquire thanks to social networks or other factors.

Currently, the development of this research is planned to improve the user experience by:

– including in the recipe database (Fig. 4) data on ready-made meals from supermarkets, restaurants, fast food chains, and delivery. These meals are popular and are an integral part of the everyday life of many people. Including them in the planning system will make it more accurate and efficient and will also allow them to better reflect real life;

– taking into account the average prices of ingredients and ready-made meals in order to balance the total cost of food for a given period.

Among other improvements planned in the future:

– adding cultural and demographic characteristics of the user;

– taking into account the seasonality of user preferences;

– adding a contextual component that will allow taking into account trends.

However, most of these improvements also involve adding new data, which is significantly limited by the resource component and faces the difficulties of collecting experimental data for training models.

7. Conclusions

1. The architecture of an intelligent system for personalized planning of a balanced diet and separate training of recommendation systems for each meal, focused on the individual characteristics of the user, has been proposed. It is shown that the use of effective separately trained recommendation systems for breakfast, lunch, snack, and dinner menus provides the system with flexible menu generation due to training on data sets containing only recipes for the corresponding food categories. High adaptability to user needs is guaranteed by using the Precision@K metric to assess the effectiveness of recommendation systems during training.

2. An experimental study of candidate models for recommendation systems for each type of meal has been performed. At the same time, separate training of recommendation systems for each meal allowed us to investigate the effectiveness of different models for breakfast, lunch, snack, and dinner and made it possible to better adapt the models to the user's needs and the specificity of each type of meal. Specifically, for the SVD model, the improvement in Precision@10 for specialized configurations trained for breakfast, lunch, snacks, and dinner compared to the universal model trained on the entire dataset was 10 to 18 percent. For GBDT, this improvement was 17 to 21 percent, and for Wide&Deep, it was 18 to 23 percent.

3. Analysis of the results of our experimental studies revealed the feasibility of using the Wide&Deep model for lunches, snacks, and dinners, and the SVD model for breakfast recommendations. In particular, according to the Precision@10 metric:

– SVD showed the best result for breakfasts (~94 % of relevant recommendations);

– Wide&Deep showed the highest efficiency (~96 % of relevant recommendations for lunches, ~97 % for snacks, and ~95 % for dinners).

These results indicate the potential of the proposed approach to significantly improve existing meal planning systems by using different models optimally tuned for each meal.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

The data will be provided upon reasonable request.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Overweight and obesity – BMI statistics. Eurostat. Available at: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Overweight_and_obesity_-_BMI_statistics
2. Misra, A., Jayawardena, R., Anoop, S. (2019). Obesity in South Asia: Phenotype, Morbidities, and Mitigation. Current Obesity Reports, 8 (1), 43–52. https://doi.org/10.1007/s13679-019-0328-0
3. Powell-Wiley, T. M., Poirier, P., Burke, L. E., Després, J.-P., Gordon-Larsen, P., Lavie, C. J. et al. (2021). Obesity and Cardiovascular Disease: A Scientific Statement From the American Heart Association. Circulation, 143 (21). https://doi.org/10.1161/cir.0000000000000973
4. Adair, T., Lopez, A. D. (2020). The role of overweight and obesity in adverse cardiovascular disease mortality trends: an analysis of multiple cause of death data from Australia and the USA. BMC Medicine, 18 (1). https://doi.org/10.1186/s12916-020-01666-y
5. Liu, J., Rehm, C. D., Onopa, J., Mozaffarian, D. (2020). Trends in Diet Quality Among Youth in the United States, 1999-2016. JAMA, 323 (12), 1161. https://doi.org/10.1001/jama.2020.0878

6. Pecune, F., Callebert, L., Marsella, S. (2020). A Socially-Aware Conversational Recommender System for Personalized Recipe Recommendations. Proceedings of the 8th International Conference on Human-Agent Interaction, 78–86. https://doi.org/10.1145/3406499.3415079
7. Papastratis, I., Konstantinidis, D., Daras, P., Dimitropoulos, K. (2024). AI nutrition recommendation using a deep generative model and ChatGPT. Scientific Reports, 14 (1). https://doi.org/10.1038/s41598-024-65438-x
8. Stefanidis, K., Tsatsou, D., Konstantinidis, D., Gymnopoulos, L., Daras, P., Wilson-Barnes, S. et al. (2022). PROTEIN AI Advisor: A Knowledge-Based Recommendation Framework Using Expert-Validated Meals for Healthy Diets. Nutrients, 14 (20), 4435. https://doi.org/10.3390/nu14204435
9. Zioutos, K., Kondylakis, H., Stefanidis, K. (2023). Healthy Personalized Recipe Recommendations for Weekly Meal Planning. Computers, 13 (1), 1. https://doi.org/10.3390/computers13010001
10. Majumder, B. P., Li, S., Ni, J., McAuley, J. (2019). Generating Personalized Recipes from Historical User Preferences. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 5975–5981. https://doi.org/10.18653/v1/d19-1613
11. Mohan, A., Singh, S., Kalpana, Dr. A. V. (2023). Meal Plan Monitoring and Recommendation System. Recent Trends in Data Science and Its Applications, 595–602. https://doi.org/10.13052/rp-9788770040723.117
12. Zhang, S., Yao, L., Sun, A., Tay, Y. (2019). Deep Learning Based Recommender System. ACM Computing Surveys, 52 (1), 1–38. https://doi.org/10.1145/3285029
13. Ladyzhets, V., Yeremenko, B., Terenchuk, S. (2024). Candidate Generation for Meal Recommendation System. 2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST), 560–564. https://doi.org/10.1109/sist61555.2024.10629517
14. Pu, L., Faltings, B. (2013). Understanding and improving relational matrix factorization in recommender systems. Proceedings of the 7th ACM Conference on Recommender Systems, 41–48. https://doi.org/10.1145/2507157.2507178
15. Li, D., Jin, R., Gao, J., Liu, Z. (2020). On Sampling Top-K Recommendation Evaluation. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. https://doi.org/10.1145/3394486.3403262
16. Manety, S., Khider, D., Heiser, C., McKay, N., Emile-Geay, J., Routson, C. (2022). PaleoRec: A sequential recommender system for the annotation of paleoclimate datasets. Environmental Data Science, 1. https://doi.org/10.1017/eds.2022.3
17. Hug, N. (2020). Surprise: A Python library for recommender systems. Journal of Open Source Software, 5 (52), 2174. https://doi.org/10.21105/joss.02174
18. Guolin, K. et al. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. 31st Conference on Neural Information Processing Systems. Available at: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf
19. Abadi, M. et al. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv. https://doi.org/10.48550/arXiv.1603.04467
20. Jiao, J., Zhang, X., Li, F., Wang, Y. (2020). A Novel Learning Rate Function and Its Application on the SVD++ Recommendation Algorithm. IEEE Access, 8, 14112–14122. https://doi.org/10.1109/access.2019.2960523
21. Mat Amin, M., Yep Ai Lan, J., Makhtar, M., Rasid Mamat, A. (2018). A Decision Tree Based Recommender System for Backpackers Accommodations. International Journal of Engineering & Technology, 7 (2.15), 45. https://doi.org/10.14419/ijet.v7i2.15.11210
22. Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H. et al. (2016). Wide & Deep Learning for Recommender Systems. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. https://doi.org/10.1145/2988450.2988454