SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE

PROJECT PROPOSAL

ON

DIET RECOMMENDED SYSTEM

PROJECT WORK SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF MSc. (COMPUTER SCIENCE)

SUBMITTED BY

Rohan Thakur

PROJECT GUIDE

Asst. Professor Manasvi Sharma

SIES (NERUL) COLLEGE OF ARTS SCIENCE AND COMMERECE

NAAC ACCERDITED ‘A’ GRADE COLLEGE

(ISO 9001:2015 CERTIFIED INSTITUTION)

NERUL, NAVI MUMBAI - 400706

Certificate

THIS IS TO CERTIFY THAT THE PROJECT TITLED

DIET RECOMMANDED SYSTEM

IS UNDERTAKEN BY

ROHAN KIRAN THAKUR

Seat No: __20______________________ –

In partial fulfilment of MSc - IT / CS Degree (Semester _ **III** _ Examination in the academic year **2021-2022** and has not been submitted for any other examination and does not form part of any other course undergone by the candidate. It is further certified that he/she has completed all the required phases of the project.

Project Guide External Examiner

Head of Department Principal

ACKNOWLEDGMENT

I extend my heartfelt gratitude and thanks to Asst. Professor Mansvi Sharma sir for providing me excellent guidance to work on this project and for their understanding and assistance by providing all the necessary information needed for my project topic.

I would also like to acknowledge all the staffs for providing a helping hand to us in times of queries & problems. The project is a result of the efforts of all the peoples who are associated with the project directly or indirectly, who helped us to work to complete the project within the specified time frame. They motivated me in the project and gave a feedback on it to improve my adroitness.
Thanks to all my teachers, who were a part of the project in numerous ways and for the help and inspiration they extended to me and for providing the needed motivation.
With all Respects & Gratitude, I would like to thanks to all the people, who have helped for the development of the Project.

ROHAN THAKUR

MSc. Computer Science (Part-II)

SIES (Nerul) College of Arts, Science, and Commerce.

Table of Content

Project On:

Diet Recommended System

Introduction:

Recommender systems (RS) suggest items of interest to users of information systems or e-business systems and have evolved in recent decades. A typical and well-known example is Amazon's suggest service for products. We believe the idea behind recommender systems can be adapted to cope with the special requirements of the health domain. Recommendations based on single foods or food groups are easier to implement when only a few foods serve as the major sources of an essential dietary component (e.g., dairy products, which are the primary source of calcium in the Indian diet).

During the last decades huge amounts of data have been collected in clinical databases representing patients' health states (e.g., as laboratory results, treatment plans, medical reports, diet plans). Hence, digital information available for patient-oriented decision making has increased drastically but is often scattered across different sites. As a solution, personal diet recommender systems (DRS) are meant to centralize an individual's health data and to allow access for the owner as well as for authorized health professionals

Nutrients are Essential compounds that the body can't make or can't make insufficient quantity. According to the World Health Organization (WHO), these nutrients must come from food, and they're vital for disease prevention, growth, and good health. Macronutrients are eaten in large amounts and include the primary building blocks of your diet protein, carbohydrates, and fat which provide your body with energy. Vitamins and minerals are micronutrients, and small doses go a long way. Most of disease occurred due to efficiency of nutrients. To fill these nutrients, we can suggest natural diet (that have no side effects), and precautions to user.

Nutrient-based food recommendations (e.g., due to vitamin D deficiency increase the risk skin disease to reduce this we suggest oily fish, meat etc.) might be easy for public health personnel to interpret and implement, recommendations pertaining to nutrient intake would usually need to be translated by professionals into guidance about food choices for the public.

Nutrient-based recommendations must be derived often from the epidemiologic (Distribution of health-related data) data on dietary patterns. For example, the statement that diets with a high plant food and low-fat content are associated with reduced rates of certain cancers more accurately reflects present knowledge than do conclusions that diets high in selenium or isothiocyanates are likely to reduce cancer risk. The latter requires an inference about cause and effect that is not yet justified by the data.

Similar work: Electronic Medical Records (EMRs) and Electronic Health Records (EHRs) provide the technology for the electronic storage of medical data and enables hospitals and other healthcare players to share such data electronically among authorized caregivers.

Related Work:

Several works have been proposed for different recommendation systems related to diet and food. These systems are used for food recommendations, menu recommendations, diet plan recommendations, health recommendations for specific diseases, and recipe recommendations. Majority of these recommendation systems extract users' preferences from different sources like users' ratings.

A Food Recommendation System (FRS) [1] is proposed for diabetic patients that used K-mean clustering and Self Organizing Map for clustering analysis of food. The proposed system recommends the substituted foods according to nutrition and food parameters. However, FRS does not adequately address the disease level issue because the level of diabetes may vary hourly in different situations of the patient and the food recommendations may also vary accordingly.

The core algorithm implemented in our food recommender system that is used to predict the individual user's ratings for different recipes. In order to suggest useful recipes to the user, as we have described above, the system first needs to collect information about her preferences. Either during the general preference elicitation step, or after a particular recipe is tried, the user has the option to tell the system if the recipe fits her tastes. In our system, these user's preferences are collected and represented as ratings, i.e., numerical evaluations ranging from 1 to 5, and the goal of the recommendation algorithm is to predict the ratings for recipes that is worth to cook. Those recipes with highest predicted ratings are the most likely to be useful and are therefore suggested to the user. The recommended recipes can be the ones that have never been tried by the user or the ones that are worth to cook again. In this work, we have implemented an algorithm that is capable of exploiting tagging information in the rating prediction process [9]. This algorithm builds upon the popular matrix factorization method [15], in which each user u is associated with a parameters' vector $p_u \in R^k$ that models her latent preferences. Likewise, to each recipe m corresponds a vector q_m

∈ R k that models its latent properties. In this method, ratings are estimated by computing the dot product of the vectors:

$$\hat{r}_{um} = \mathbf{p}_u^T \mathbf{q}_m$$

The algorithm proposed in extends the matrix factorization method by including additional parameters used for modelling the dependencies between assigned tags and ratings. In addition to ratings, in our food recommender system users can also specify their preferences by means of tags, and the tags' valences (positive or negative), assigned to the recipes. For instance, a user may use the tag spicy to tell the system she prefers that type of recipes. Similarly, an item can be described with the tag tomato if it is one of its ingredients. We conjectured those tags assigned to an item are correlated to the ratings, and can be used to better predict user's interests. Thus, we have decided to adopt the rating prediction model described in [9] and have introduced additional feature vectors xt ∈ R k and ys ∈ R k for user's and recipe's tags, respectively. Ratings are now estimated as follows:

$$\hat{r}_{um} = \left(\mathbf{p}_u + \frac{1}{|T_u|}\sum_{t\in T_u}\mathbf{x}_t\right)^T \left(\mathbf{q}_m + \frac{1}{|T_m|}\sum_{s\in T_m}\mathbf{y}_s\right) \quad (2)$$

In the previous formula, Tu is the set of tags assigned by the user u to any recipe, and Tm is the set of tags assigned to the recipe m by any user. If Tu = ∅ or Tm = ∅ then we ignore the corresponding terms, so that in the case that Tu = Tm = ∅ the algorithm behaves exactly as standard matrix factorization. We note that by introducing latent vectors for users and recipes separately, the algorithm is able to compute predictions even if the user did not assign any tag to the recipe. When incorporating the tags into the model, we have used only the positive tags from the user. This makes the effect of the tags on rating prediction easier to isolate. In a future work, when the rating data set will also be larger, we will study how to integrate the negative tags. The parameters pu, qm, xt, ys are automatically learned during the training phase of the algorithm, by minimizing the regularized squared prediction error

$$\min_{p_*,q_*,x_*,y_*} \sum_{(u,m)\in\mathcal{R}} (r_{um} - \hat{r}_{um})^2 + \lambda\left(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_m\|^2 + \sum_{t\in T_u}\|\mathbf{x}_t\|^2 + \sum_{s\in T_m}\|\mathbf{y}_s\|^2\right) \quad (3)$$

where R is the training set, and the parameter λ ∈ R controls the complexity of the model and is used to prevent overfitting. In our experiments, we found λ = 0.02 to be a good trade-off. The previous minimization problem can be efficiently solved using stochastic gradient descent, as depicted in Algorithm 1.

Algorithm 1 Recommendation model training step

```
procedure TRAINSTEP
  for (u, m) ∈ R do
    e_um ← r_um − r̂_um                ▷ Compute r̂_um using eq. 2
    ▷ Simultaneously update the parameter vectors:
    p_u ← p_u − α [λp_u − e_um (q_m + |T_m|^-1 Σ_{s∈T_m} y_s)]
    q_m ← q_m − α [λq_m − e_um (p_u + |T_u|^-1 Σ_{t∈T_u} x_t)]
    for all t ∈ T_u do
      x_t ← x_t − α [λx_t − (e_um / |T_u|) (q_m + |T_m|^-1 Σ_{s∈T_m} y_s)]
    end for
    for all s ∈ T_m do
      y_s ← y_s − α [λy_s − (e_um / |T_m|) (p_u + |T_u|^-1 Σ_{t∈T_u} x_t)]
    end for
  end for
end procedure
```

A single training step iterates over the whole training set updating the model parameters each time. This procedure is typically repeated for a fixed number of iterations, or until convergence is reached. The parameter α controls to what extent are the parameters updated on each step: a too small value can make the learning very slow, whereas with large values the algorithm may fail to converge. In our experiments, we found good results using α = 2 · 10−4 , a single training iteration. We also found that a value of k = 5 for the dimensionality of the latent vectors offered a good trade-off between prediction accuracy and training time. Once the training phase is complete, we use the learned parameters to compute rating predictions using equation 2. In order to provide the target user with recommendations, the system first estimates her ratings for the unknown recipes, sorts them from highest to lowest predicted value, and presents the top ranked items.

The Nutritional Recommendation Approach Integrating Nutritional and User Preferences-Related Information.

A. Initial Data Preparation

The initial steps necessary to prepare the data to be used in the recommendation generation are based on two goals:

I. the construction of the food profiles
II. the definition of menu templates to be filled by the food items.

1) Construction of the Food Profiles

The food profile definition is built by taken as base two popular food composition tables provided by Wander [18]. These tables contain nutritional information of 600+ foods, related to the number of calories and 20+ different macronutrients and micronutrients. The mentioned tables arrange the foods into 12 groups, which are milks, eggs, meat, fish, leguminous, oleaginous dry fruits, oils, cereals, desserts, vegetables, fruits, and drinks. Furthermore, the tables reflect the number of calories, macronutrients, and micronutrients, in 100 g of each food. In order to make these data suitable for recommendation generation, a nutritionist determined reasonable portions for each food according to its type and features; and therefore, calculates the amount of macro and micronutrients belonging to each portion. Table 2 presents a fragment of these final data, that is the source to be used in the food profiles.

TABLE 2 Fragment of the Food Composition Tables

Food	Kilocalories	Proteins	Carbohydrates	Lipids	Cholesterol	Iron	Calcium	...
Pork chop (60 grs)	198	9	0	18	43.2	1.5	4.8	...
Rabbit (125 grs)	202.5	27.5	0	10	81.25	1.25	25	...
White rice (130 grs)	460.2	9.88	100.1	2.21	0	1.04	13	...
Lettuce (200 grs)	36	2.4	4.8	0.4	0	1.30	124	...
Guava (30 grs)	10.5	0.27	2.01	0.15	0	0.225	5.1	...
...	...	...	...	...	...	...	...	...

In this way, the foods' profiles will be composed of the amount of nutrients which have been considered as key features for characterizing foods. These nutrients are proteins, lipids, carbohydrates, cholesterol, sodium, and saturated fats; leaving to the next future works the use of a food profile considering further nutrients. Kilocalories are also discarded because its value can be calculated through the carbohydrates, proteins, and lipids values.

$$a_k = (pro_k, lip_k, cb_k, ch_k, sod_k, sat_k)$$

Furthermore, in the current work this context will be treated as a *decision table*, where the foods to be consumed are the *alternatives* and the calories and nutrients are the *decision criteria*. Table 3 formalizes the notation that will be used in the remaining of the paper, to refer to the food profile components.

TABLE 3 Criteria for Characterizing Foods

Term	Nutrient
pro_k	Amount of proteins of food k
lip_k	Amount of lipids of food k
cb_k	Amount of carbohydrates of food k
ch_k	Amount of cholesterol of food k
sod_k	Amount of sodium of food k
sat_k	Amount of saturated fats of food k

2) Definition of the Menu Templates

On the other hand, it is also necessary as initial data the definition of menu templates that will be used in the menu recommendation. A menu template follows the common scheme of a typical daily meal, and it is also built through the support of a nutrition domain expert. This menu template is composed of a breakfast, a lunch, and a dinner. In this paper we will not consider snacks, although the proposal could be easily extended to cope with them.

In order to facilitate the template definition and taking as basis the nutritionist knowledge, we group the food profiles into new groups according to their main associated nutrient and related features.

TABLE 4 New Food Groups for the Menu Generation

Group name	Group composition
Group G_1 (Milk)	Milk, yogurts
Group G_2 (Breakfast cereals)	Some cereals (e.g. bread, wheat)
Group G_3 (Sources of proteins)	Eggs, Meat, Fish
Group G_4 (Sources of carbohydrates)	Some cereals (e.g. rice), Leguminous
Group G_5 (Vegetables)	Vegetables
Group G_6 (Fruits)	Fruits

Starting from these groups, Table 5 shows the template proposed for a daily meal plan. Specifically, the values for parameters nG1,nG2,… will be proposed later in the

TABLE 5 The Template for the Daily Meal Plan

Breakfast
n_{G_1} foods of group G_1 (Milk, yogurts)
n_{G_2} foods of group G_2 (Breakfast cereals)
n_{G_6} foods of group G_6 (Fruits)
Lunch
$n^l_{G_3}$ foods of group G_3 (Proteins)
$n^l_{G_4}$ foods of group G_4 (Carbohydrates)
$n^l_{G_5}$ foods of group G_5 (Vegetables)
n_{G_6} foods of group G_6(Fruits)
Dinner
$n^d_{G_3}$ foods of group G_3 (Proteins)
$n^d_{G_4}$ foods of group G_4 (Carbohydrates)
$n^d_{G_5}$ foods of group G_5 (Vegetables)
n_{G_6} foods of group G_6(Fruits)

Content based food recommender system [3] is proposed which recommend food recipes according to the preferences already given by the user. The preferred recipes of the user are fragmented into ingredients which are assigned ratings according to the stored users' preferences. The recipes with the matching ingredient are recommended. The authors do not consider the nutrition factors and the balance in the diet. Moreover, chances of identical recommendation are also present because the preference of the user may not change on daily basis. The above-mentioned diet recommendation systems are specifically dealing with some diseases or related to balance the diet plans. In case of food recommendation for specific diseases, the systems recommend different foods for patients without knowing the level of disease which may vary in different cases and cause severe effects on patients. Similarly, in case of food recommendations to balance the diet, nutrition factors are ignored which are very much important to recommend food and balance diet.

Methodology Seeing that the requirements of the application were clear, the waterfall model was used to develop the application.
Algorithm used Content Based Filtering Algorithm**:** In a content-based recommender system, keywords or attributes are used to describe items. Calculate the weight of each feature, namely Calories, that has the lowest value. This step is in order to make an accurate list of food items. If the category **i** has the lowest value **Cj,** but it has many contents' features **j** in specific category **i ,** the application needs to compute the weight Wi of each feature content value Ci within that category. The equation given below is adopted from below.

$$W_i = \frac{C_i * \sum_{j=0}^{n} C_j}{n}$$

After computing the weight of each feature content, the highest weight is seen as having the lowest value. Only the food items which have the lowest calories value is shown in the list to the user. The list is sorted in value of the calories from lowest to highest

Objectives:

The aim of this project was to build a food recommendation system for disease. This included to pre-process nutrition-based food and nutrition-based disease dataset for diet retrieval. To optimize the vocabulary of food to match them in the disease. To train, evaluate and test a K means clustering model and Random Forest able to predict the Food that disease belongs, by considering its set of nutrition. Predict the probability food according to disease

This project study is to consider various important aspects of the user's lifestyle and make sure that these factors are incorporated while the system works on a solution to build and recommend a healthy and nutritious diet for the user. A good nutritious healthy diet and a moderate amount of physical activity can help in maintaining a healthy weight. But the benefits of good nutrition have a lot more to do than just managing the weight. Being fit is all about the 70/30 rule. Here’s how it goes, for a person to stay healthy he/she must focus 70% on his dietary intake and 30% on his physical activity/exercise.

DRS is project to provide users with healthy diet and individual nutritional recommendation. Health dietary was based on user disease. Furthermore, DRS contains catalogues of typical foods from an Asian region. Several Calabrian foods have been inserted because of their nutraceutical properties widely reported in several quality studies. DRS includes disease-based precautions.

Methodology:

Research methodology is a process that includes a number of activities to be performed. These are then arranged in proper sequence for conducting research. It is a master plan specifying the method and procedures for collecting and analysing needed information. Descriptive Research is used in this study as the main aim is to describe characteristics of the phenomenon or a situation. As the issue is well understood, it focuses on the development of in-depth knowledge the facts will be used to analyse and evaluate the data.

How it will be done:

The Data:

- Collecting dataset: I have found a dataset from Kaggle also I am going to collect the data from WHO.
- I will have to manually collect data using web scrapping.

Data Pre-Processing:

- At this stage removing punctuation, stop-words, special characters, making the table heading lowercase etc. will be done.
- Convert Integer values into Float.
- Null values in dataset can be removed or filled.

Splitting of the dataset:

- The dataset will be divided into training set and testing set.
- My train dataset size(population) is 6692 so my test (sample) data size will be 364 with margin of error is 5%.

Topic Modelling:

- Performance Evaluation: Here the models used will be evaluated using evaluation techniques like f1- score, accuracy, etc.
- Complaints:
    - I. I am use normal diet as required to normal Human.
    - II. If disease become detected serious then that become excluded.
    - III. False-Negative result will be included.

Implementation:

1. User's will enter the necessary information like their disease name, weight etc. on the form.
2. The information will then go through the ML model in following manner:
   i. **K-Means** is used for clustering to cluster the food according to nutrition's.
   ii. **Random Forest Classifier** is used to classify the food items and predict the food items based on input
3. The System will then recommend diet to the users based on input
4. The Users can choose from multiple recommended items and make their diet plan.

Classification Model:

We use the training dataset to get better boundary conditions that could be used to determine each target class. Once the boundary conditions are determined, the next task is to predict the target class. The whole process is known as classification. Classification will be based on various factors that are quality, quantity, search process and payment etc. To find out answers of above questions, here I am using some algorithms base on that I can easily classify my data.

Random Forests and k-Means falls under different categories of algorithms. k-Means is an unsupervised clustering algorithm wherein you group/cluster records. And Random Forests is a supervised learning algorithm used generally for classification and regression problems.

K Means Clustering:

K-Means clustering is an **unsupervised learning algorithm**. There is no labelled data for this clustering, unlike in supervised learning. we often use classification or regression algorithms in supervised learning methods to predict categories or values, we still often encounter situations where we need to use unsupervised learning methods to obtain a set of data categories. When the amount of data is large, you can consider using clustering algorithms to get different data categories. Clustering is subordinate to unsupervised learning, which does not rely on the defined classes and training examples of class labels. Among them, K-means clustering is a very classic clustering method [8]

Given a set of elements, where each element has observable attributes, use a certain algorithm to divide into subsets, and require the degree of difference between the elements within each subset as much as possible low, and the element dissimilarity of different subsets is as high as possible. Concentration, each subset is called a cluster. Different from classification, classification is exemplary learning, which requires that each category be clarified before classification and that each element is mapped to a category, while clustering is observational learning, and the category may not be known or even the number of categories may not be known before clustering.

K-means tries to find the natural category of the data. The user sets the number of categories to find a good category centre. The algorithm flow is as follows:

(1) Enter the number of data sets and categories K
(2) Randomly assign the centre point of the category
(3) Put each point into the set of the category centre point closest to it
(4) Move the category centre point to the set where it is
(5) Go to step 3 until convergence

After a number of cycles, the best classification effect can be obtained. Different from marine shale reservoirs, the relationship between food content of coal reservoirs and nutrition's of the coal reservoirs is relatively poor, and the laws are inconsistent, which also leads to the unreliability of the final prediction model. This is because coal reservoirs are more complex than shale reservoirs and have worse continuity, which causes the logging of coal seams to be affected by multiple factors. Using the clustering method to obtain multiple categories and establishing corresponding prediction models based on different categories can greatly improve the prediction results.

Random Forest Algorithm:

Random forest is a **supervised learning algorithm**. The "forest" it builds, is an ensemble of decision trees, usually trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. Random forest is a highly flexible machine learning algorithm that has just emerged in the 21st century. It refers to a classifier that contains multiple decision trees. The thinking behind it is similar to group wisdom. In the 1980s, Breiman et al. invented an algorithm for classification trees, which performed classification or regression through repeated dichotomy of data, which greatly reduced the amount of calculation. In 2001, Breiman combined the classification trees into a random forest, that is, randomized the use of variables and the use of data, generated many classification trees, and then summarized the results of the classification trees [8].

Random forest improves the prediction accuracy without a significant increase in the amount of calculation. Random forest is not sensitive to multivariate collinearity, and the results are relatively robust to missing data and unbalanced data and can well predict the effect of thousands of explanatory variables.

Random forest uses a random method to build a forest. There are many decision trees in the forest, and there is no correlation between each decision tree in the random forest. After obtaining the forest, when a new input sample enters, let each decision tree in the forest make a judgment separately to see which category the sample belongs to. The class with the most classification times is the predicted class. Random forest can handle quantities whose attributes are discrete values. The construction process of random forest is as follows:

(1) If there are **N** samples, samples are randomly selected for replacement (one sample is randomly selected each time and then returned to continue selection). Use the selected **N** samples to train a decision tree as the sample at the root node of the decision tree

(2) When each sample has **M** attributes, when each node of the decision tree needs to be split, then **m** attributes are selected from these **M** attributes, and the condition **m<M** is satisfied. Then, from these **m** attributes, strategies such as information gain are used to select one attribute as the split attribute of the node

(3) In the process of decision tree formation, each node must be split according to step 2 until it can no longer be split. Note that there is no pruning during the entire decision tree formation process.
(4) Follow steps 1-3 to build a large number of decision trees to form a random forest

In the process of building each decision tree, attention should be paid to the impact of sampling and complete splitting. The first is two random sampling processes. Random forest samples the input data in rows and columns. For line sampling, a replacement method is used, that is, in the sample set obtained by sampling, there may be duplicate samples. Assuming that there are **N** input samples, there are also **N** samples sampled. In this way, when training, the input samples of each tree are not all samples, making it relatively difficult to overfitting. Then, perform column sampling, from **M** features, select **m(m<M).**
After that, a decision tree is built using a completely split method for the sampled data, so that a certain leaf node of the decision tree cannot continue to split, or all the samples in it point to the same category. Generally, many decision tree algorithms have an important step-pruning, but this is not done here. Since the previous two random sampling processes ensure randomness, even if pruning is not performed, overfitting will not occur. Using a random forest method to predict food content should be able to achieve better results.

Combination Method of K-Means Clustering and Random Forest:

It is difficult to evaluate the food content of nutrition’s, because the result has been affected by various factors, resulting in a poor relationship between food content and nutrition’s. Only by using clustering and other methods to truly combine nutrition’s for classification, different types of data are affected differently, and the relationship between nutrition’s and food content in different categories is closer.

Therefore, K-means clustering is performed first, and then based on the results of the clustering, a random forest model of different types is established for final application. In fact, the inherent meaning of this model is similar to that of random forests. It uses K-means clustering combined with random forests to form a “forest group” to predict food content more accurately. The modelling and forecasting process is as follows:

i. Use K-means clustering to divide the data into several categories. The measurement method usually used to compare the results of different **K** values is the average distance between a data point and its cluster centroid. Since increasing the number of clusters will always reduce the distance to the data point, when is the same as the number of data points, increasing **K** will always reduce the metric to zero. Therefore, this indicator cannot be used as the sole target. Conversely, the average distance to the centre of mass is plotted as a function of **K**, and the “elbow point” at which the reduction rate changes sharply can be used to roughly determine the **K** value.

ii. Use **K** sets of data and random forest algorithm to train **K** models. After determining the category of the new data, the corresponding model can be used to calculate the food content.

iii. When predicting new data, first determine the category of the new data by calculating the Euclidean distance between the sample data and the centroids of multiple classes of data. The new data belongs to the category corresponding to the centroid with the smallest Euclidean distance. After the category is determined, the corresponding model is used for prediction, and the predicted value of the food content of the sample point is obtained, and the reliability of the algorithm is determined by comparing with the real value

Requirements:

The experiment would be implemented completely in python language specification of it may be as follows:

• Python 3.5 and above
• Python IDE and Visual Code will be used as package manager for python environment.
• Google Collab will be used as online python environment.

Conclusion:

This project satisfying need will help to put patients in control of their own health data and therefore increase patients' autonomy. An approach of integrating recommender systems into personal Diet recommender system (DRS) was outlined. we can suggest natural diet (that have no side effects), precautions to user. The proposed system builds a user's health profile and, accordingly, provides individualized nutritional recommendations, also with attention to food geographical origin. The importance of nutritional guidance is increasing day by day to lead a healthy and fit life and by accepting the user's preferences and a user's profile in the system a healthy diet plan is generated.

Reference:

1. https://ieeexplore.ieee.org/document/8765311
2. Ge, M., Elahi, M., Fernaández-Tobías, I., Ricci, F., & Massimo, D., "Using tags and latent factors in a food recommender system," in Proc. of the 5th International Conference on Digital Health, pp. 105-112, ACM., May 2015.
3. Freyne, J., & Berkovsky, S., "Evaluating recommender systems for supportive technologies," User Modeling and Adaptation for Daily Routines, pp. 195-217, Springer London, 2013.
4. https://www.sciencedirect.com/science/article/abs/pii/S0169260716306927#:~:text=DIETOS%20is%20a%20novel%20food,attention%20to%20food%20geographical%20origin.
5. Recommender systems in the health care domain: state-of-the-artandresearchissues by ThiNgocTrangTran1 ·AlexanderFelfernig1·ChristophTrattner2· AndreasHolzinger3
6. Atas, M., Tran, T.N.T., Felfernig, A., Polat-Erdeniz, S., Samer, R., Stettinger, M. (2019). Towards similarity aware constraint-based recommendation. In Advances and trends in artificial intelligence, lecture notes in computer science, Springer.
7. Bankhele,S.,Mhaske,A.,Bhat,S.(2017).V.,s.:adiabetichealthcarerecommendationsystem.International Journal of Computer Applications,
8. https://www.hindawi.com/journals/geofluids/2021/9321565/