Recommender System on Nutritional Values for Different Food Items

Anand(anand6@buffalo.edu), Rahul Patil(rpatil@buffalo.edu)
Masters in Data Science,
School of Engineering and Applied Sciences
University at Buffalo
Buffalo, NY, 14214

Abstract

The vast amount of data available today allows information enthusiasts with right skills to draw fruitful insights by inferences and patterns from data. These inferences and patterns give rise to the concept of Recommender Systems which play a major role in our day-to-day decisions in the form of various categories like "Recommended Items", "You may also Like" or "People you may know" in fields of ecommerce and social media. We intend to extend the capabilities of Recommender system to find similarities and diversity in various food nutritional values and present them as alternative to the food being looked for.

I. Introduction

COVID-19 is an unforeseen pandemic that struck the world on an unprecedented scale during the early months of 2020. With panic and lot of measures being placed, one of the most essential necessities is stocking up on food items as there are social restrictions placed in various different parts of the country with little to no access to all the food items that are otherwise available easily. With the growing need to find alternative foods with same nutritional values, it seems apt to design a recommender system that would recommend various food items with similar nutritional values in same or different food types with flexibility as desired by the user.

A recommender system, or a recommendation system is a subclass of information filtering systems that seeks to predict the "Rating" or "Nearness" of an item. The goals of a recommender system are relevance to the product being searched, novelty, serendipity, Increasing recommendation diversity. There are different types of recommender systems such as collaborative recommender system, content-based recommender system, knowledge-based recommender system, demographic recommender system, Utility based recommender system, Hybrid recommender system. In this project, we use item collaborative filtering with neighborhood-based methods to make recommendations. Diversity is a term sparsely concentrated in any recommender system. The objective of this project is providing the user with an option to diversify their recommendations where we recommend them a number of popular but less similar items while keeping in mind the nutritional requirement of the food item in question.

II. Conceptual Approach

Item-based collaborative filtering is a model-based algorithm for making recommendations. In the algorithm, the similarity between different items in the dataset is calculated by using a similarity measure, and then these similarity values are used to predict nearness to the searched food item. Here, we use the method of K-Nearest Neighbors to recommend the most similar food items based parametrically on the nutritional values. The KNN algorithm assumes that similar things exist in close proximity.

The k nearest neighbor classifier is based on the Euclidian distance between products. Let x_i be the input sample with p features (x_{i1}, x_{i2}…x_{ip}), n be the total samples (I = 1,2, 3...n) and p be the total number of features (j = 1,2,3…p). Then, the Euclidian distance between x_i and x_l (l = 1, 2, 3…n) is defined as

$$d(xi, xl) = \sqrt{}(x_{i1}-x_{l1})^2 + (x_{i2}-x_{l2})^2 + \ldots + (x_{ip}-x_{lp})^2$$

The predictions of KNN are directly from the dataset. For new sample(x), the prediction is done by searching through the entire dataset for the nearest k-neighbors by using Euclidian distance. KNN is a lazy learning algorithm, which means, we do not need model fitting or a training step. The value of k in KNN determines the number of nearest neighbors to be selected.

In recommender systems, by informal definition, a longtail is divided into two separate parts: the head and the tail. The head part contains of most similar items (in this case, the items with nearest distance) which are often items or recommendations of the same food type. And the tail of the curve is characterized by the remainder of the products. Computationally, the long tail model, F(x) simulates any heavy-tailed distribution. It models the cumulative distribution of long tail data F(x) which represents the share (%) of total volume covered by objects up to rank x:

$$F(x)= \frac{\beta}{(\frac{N_M}{x})^{\alpha+1}}$$

Where α is the factor that defines S-Shape of the function. β is the total volume share and N_M is the median of the number of objects that cover half the total volume.

In this project, we diversify the items from the recommender system based on user selection by selecting the 'n' items from the tail end, to provide a different set of recommendations that are distinct in product type from remainder of items from the KNN.

III. Dataset and Preprocessing of Data

The dataset being used in "Composition of Foods, Raw, Processed, Prepared USDA National Nutrient Database for Standard Reference, Release 28" from U.S. Department of Agriculture. This dataset contains data for all food items in Standard Reference (SR-28), with nutrient values in 53 columns. These data are presented per 100 grams, edible portion. The Daily values of Nutrition and Supplement is obtained from the U.S. Food and Drug Administration.

For data preprocessing, we first scale down the number of different nutrients to most essential ones, viz. 'Energ_Kcal', 'Lipid_Tot_(g)', 'FA_Sat_(g)', 'Cholestrl_(mg)', 'Sodium_(mg)', 'Carbohydrt_(g)', 'Fiber_TD_(g)', 'Sugar_Tot_(g)', 'Protein_(g)', 'Vit_D_µg', 'Calcium_(mg)', 'Iron_(mg)', 'Potassium_(mg)'.

Then, we clear the rows with missing values of data. Since, each of the above nutrients have different measurement scales, scaling of data becomes essential. We use a Stadardscaler from python package 'sklearn' to attain scaling.

This data is then saved to a excel spreadsheet as "Preprocessed_Data.csv" for predictions.

```
#Reading the dataset
datafile= pd.read_excel("sr28abxl\ABBREV.xlsx")

#Extraction of features
features=['Energ_Kcal',
          'Lipid_Tot_(g)',
          'FA_Sat_(g)',
          'Cholestrl_(mg)',
          'Sodium_(mg)',
          'Carbohydrt_(g)',
          'Fiber_TD_(g)',
          'Sugar_Tot_(g)',
          'Protein_(g)',
          'Vit_D_µg',
          'Calcium_(mg)',
          'Iron_(mg)',
          'Potassium_(mg)']

#Filtering only the 'Features' columns from the dataset
attributes = datafile[features]

#Drop missing values
attributes = attributes.dropna()
dataset = attributes

#Scaling the cleaned data
scaled = StandardScaler()
dataset_scaled = scaled.fit_transform(attributes)

dataset["Itemname"] = datafile["Shrt_Desc"]
dataset["NDB_No"] = datafile["NDB_No"]
dataset = dataset.dropna()

#Saving Preprocessed data
dataset.to_csv("Preprocessed_dataset.csv")
np.savetxt("Scaled_dataset.csv", dataset_scaled, delimiter=",")
```

IV. KNN Model, Front-End design and Computational Approach

We used the package ‘Scikit Learn’ for the computation of K-Nearest Neighnbors. Scikit-learn is an open source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities. Sklearn.neighbors provides functionality for unsupervised and supervised neighbors-based learning methods. The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. The number of samples can be a user-defined constant (k-nearest neighbor learning).

In scikit learn, KneighborsClassifier implements learning based on the nearest neighbors of each query point, where is an integer value specified by the user. Here, we are specifying the value of k = 100, such that first 100 similar products are listed. The output of KNN is stored in file ‘finalized_model.sav’ and the Euclidian distances in file ‘finalized_model_dist.sav’.

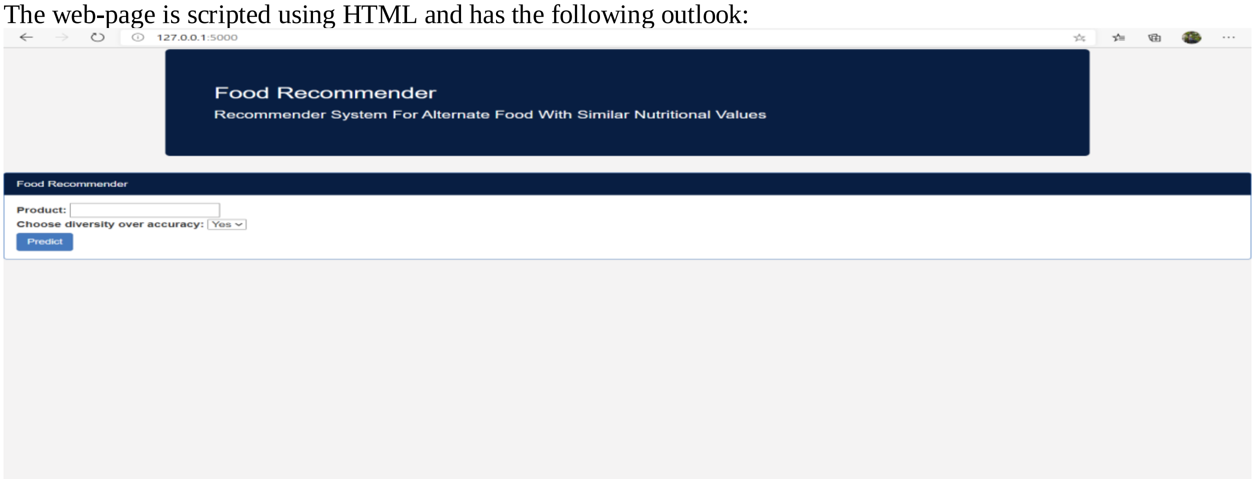
```
#Building the model using K-NN with K = 100
recommendations = NearestNeighbors(n_neighbors=100, algorithm='auto').fit(dataset_scaled)

#Euclidian Distance Calculation
distances,item_indices = recommendations.kneighbors(dataset_scaled)
```

The distance matrix generated is as follows:

```
In [2]: runcell(0, 'C:/RecommenderSystem/CodeBase/recomimpl.py')
[[0.         0.61759664 1.04453146 ... 7.23189118 7.24119297 7.25147386]
 [0.         1.04453146 1.04938804 ... 6.45908489 6.46093493 6.47629927]
 [0.         2.47958752 2.5172238  ... 9.24217831 9.24846557 9.30596899]
 ...
 [0.         0.6272778  0.63303357 ... 2.46431651 2.47899462 2.48355696]
 [0.         0.40336762 0.43586258 ... 0.76842859 0.7684736  0.76869717]
 [0.         0.53225514 0.53313326 ... 0.73063537 0.73118318 0.73243512]] [[   0  107
1 ...  582 2275   56]
 [   1    0  107 ... 4673 3445 2312]
 [   2    0  107 ... 3503 4823 4673]
 ...
 [4840 4825 3587 ...  157 3443   70]
 [4841 2804 4837 ... 4183 4266 4318]
 [4842  781  806 ... 4152 1575 3961]]
```

We use the package “flask” to build a WebApp framework for the front-end UI of the recommender.
The web-page is scripted using HTML and has the following outlook:



With the K-Nearest Neighbors calculated and Distance Matrix calculated, we use the following libraries:

Flask	Flask is a popular, extensible web microframework for building web applications with python
JobLib	JobLib is a set of tools that provide lightweigh pipelining in python
FuzzyWuzzy	FuzzyWuzzy is a library of python which is used for string matching by calculating the difference between sequences
Pandas	Pandas is an open-source data analysis and manipulation tool built on top of Python

In this program, we use the FuzzyWuzzy to introduce diversity to the system. FuzzyWuzzy package matches the string accuracy in a given sequence. When diversity is True, we use the output from FuzzyWuzzy to select the items that do not belong to a particular food type.

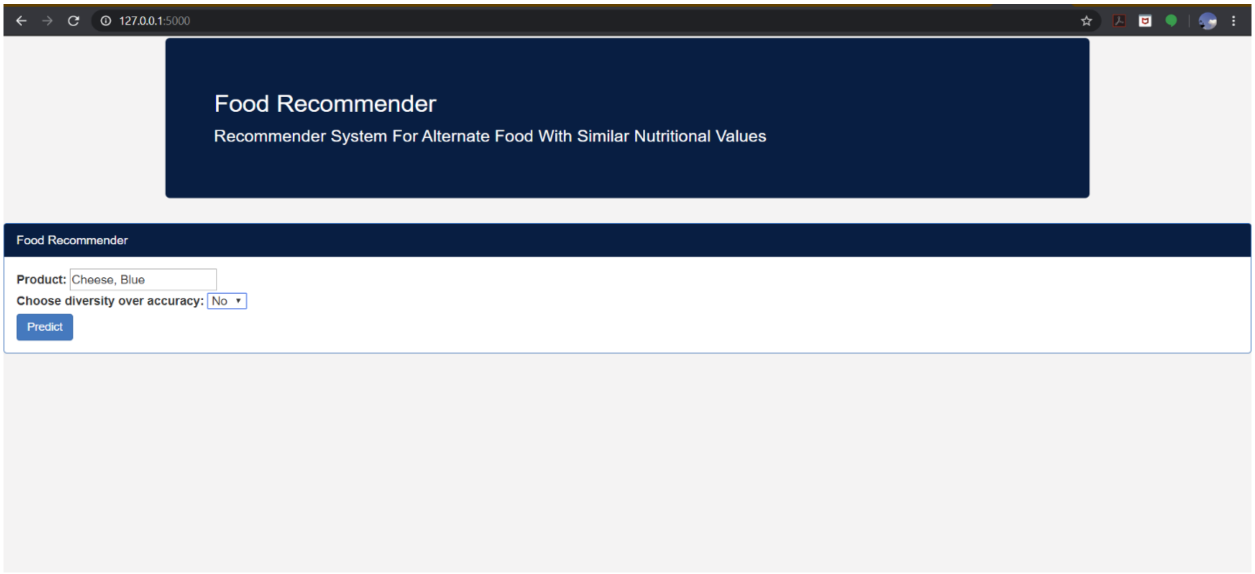
For ex. If searched for food type ‘Cheese,Blue’ and diversity is False, then the top 8 results belong to the food type ‘cheese’. But if diversity is ‘True’, the program outputs list of food items from the top 100 nearest neighbors, whose string match is less than 40%.

With the following libraries invoked, we create a function predict () in python file app.py, which follows the flowas below:

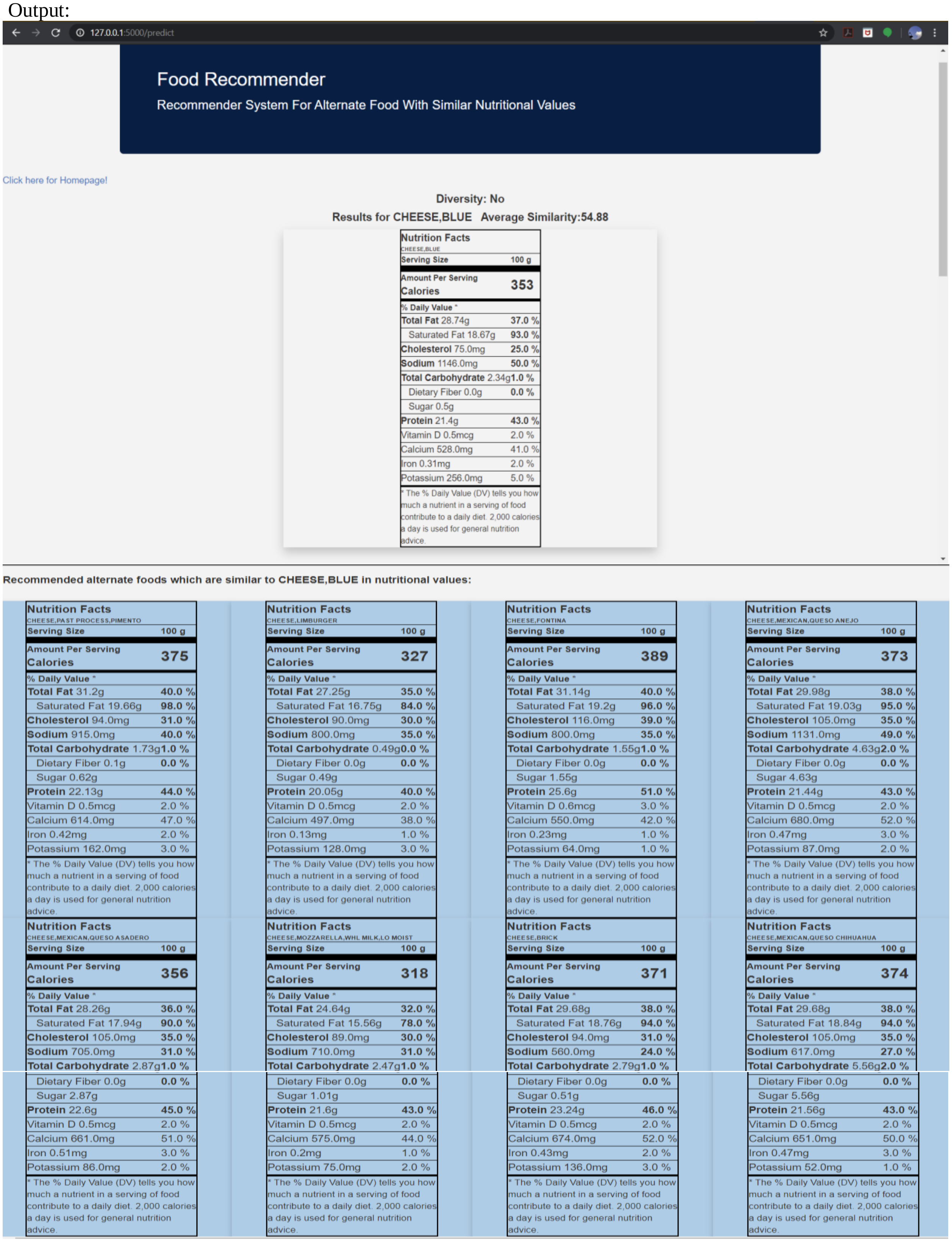
1. Load Welcome page ‘welcome.html’ to the browser
2. The values of daily nutrients are preloaded (Changes optional, if any changed)
3. Load files finalized_model.sav and finalized_model_dist.sav
4. Read data from file ‘Preprocessed_data.csv’
5. When Food item is input on the ‘welcome.html’ page, initialize empty lists for all nutrient columns
6. Read Inputs ‘Food Item’ and ‘Diversity’
7. Check if the given food name is present in the Preprocessed_dataset.csv
8. If present, Fetch the values of various nutrients from the dataset
9. If daily value is a non-zero entity, compare each of the nutrients with daily values and calculate the percentage
10. Calculate the similarity percentage
11. If input for Diversity = ‘YES’, then show the items which have accuracy of less than 40%
12. If input for Diversity = ‘NO’, show the top 8 nearest neighbor food items
13. Zip each of the nutrient and values under food items
14. Return the list in a formatted template on the ‘results.html’ page

The output page for a search for food item ‘Cheese,Blue’ is as displayed:

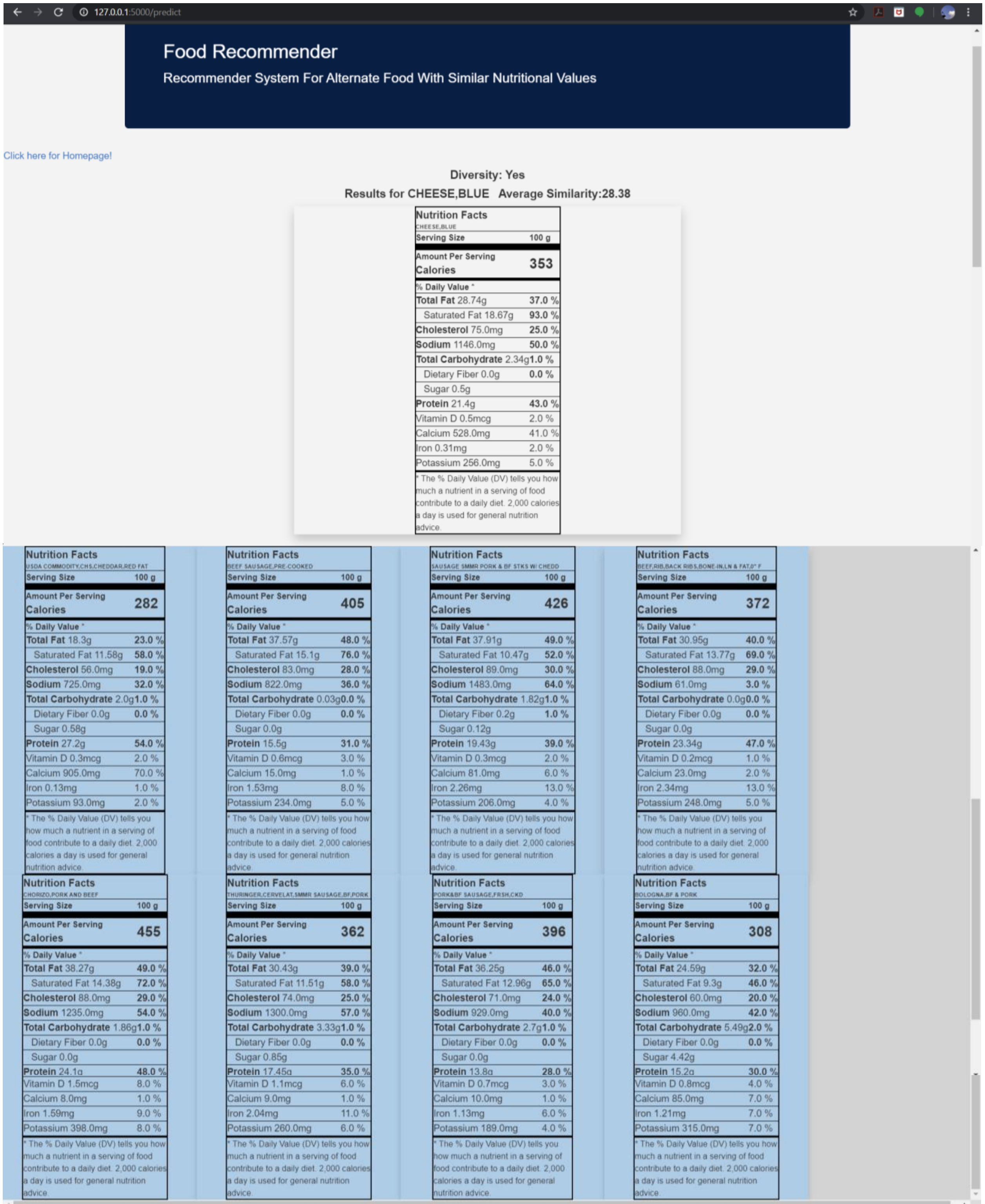
1.With Diversity = No



Output:


127.0.0.1:5000/predict
Food Recommender
Recommender System For Alternate Food With Similar Nutritional Values
Click here for Homepage!
Diversity: No
Results for CHEESE,BLUE Average Similarity:54.88
Nutrition Facts
CHEESE,BLUE
Serving Size 100 g
Amount Per Serving
Calories 353
% Daily Value *
Total Fat 28.74g 37.0 %
Saturated Fat 18.67g 93.0 %
Cholesterol 75.0mg 25.0 %
Sodium 1146.0mg 50.0 %
Total Carbohydrate 2.34g 1.0 %
Dietary Fiber 0.0g 0.0 %
Sugar 0.5g
Protein 21.4g 43.0 %
Vitamin D 0.5mcg 2.0 %
Calcium 528.0mg 41.0 %
Iron 0.31mg 2.0 %
Potassium 256.0mg 5.0 %
* The % Daily Value (DV) tells you how much a nutrient in a serving of food contribute to a daily diet. 2,000 calories a day is used for general nutrition advice.
Recommended alternate foods which are similar to CHEESE,BLUE in nutritional values:
Nutrition Facts
CHEESE,PAST PROCESS,PIMENTO
Serving Size 100 g
Amount Per Serving
Calories 375
% Daily Value *
Total Fat 31.2g 40.0 %
Saturated Fat 19.66g 98.0 %
Cholesterol 94.0mg 31.0 %
Sodium 915.0mg 40.0 %
Total Carbohydrate 1.73g 1.0 %
Dietary Fiber 0.1g 0.0 %
Sugar 0.62g
Protein 22.13g 44.0 %
Vitamin D 0.5mcg 2.0 %
Calcium 614.0mg 47.0 %
Iron 0.42mg 2.0 %
Potassium 162.0mg 3.0 %
* The % Daily Value (DV) tells you how much a nutrient in a serving of food contribute to a daily diet. 2,000 calories a day is used for general nutrition advice.
Nutrition Facts
CHEESE,LIMBURGER
Serving Size 100 g
Amount Per Serving
Calories 327
% Daily Value *
Total Fat 27.25g 35.0 %
Saturated Fat 16.75g 84.0 %
Cholesterol 90.0mg 30.0 %
Sodium 800.0mg 35.0 %
Total Carbohydrate 0.49g 0.0 %
Dietary Fiber 0.0g 0.0 %
Sugar 0.49g
Protein 20.05g 40.0 %
Vitamin D 0.5mcg 2.0 %
Calcium 497.0mg 38.0 %
Iron 0.13mg 1.0 %
Potassium 128.0mg 3.0 %
* The % Daily Value (DV) tells you how much a nutrient in a serving of food contribute to a daily diet. 2,000 calories a day is used for general nutrition advice.
Nutrition Facts
CHEESE,FONTINA
Serving Size 100 g
Amount Per Serving
Calories 389
% Daily Value *
Total Fat 31.14g 40.0 %
Saturated Fat 19.2g 96.0 %
Cholesterol 116.0mg 39.0 %
Sodium 800.0mg 35.0 %
Total Carbohydrate 1.55g 1.0 %
Dietary Fiber 0.0g 0.0 %
Sugar 1.55g
Protein 25.6g 51.0 %
Vitamin D 0.6mcg 3.0 %
Calcium 550.0mg 42.0 %
Iron 0.23mg 1.0 %
Potassium 64.0mg 1.0 %
* The % Daily Value (DV) tells you how much a nutrient in a serving of food contribute to a daily diet. 2,000 calories a day is used for general nutrition advice.
Nutrition Facts
CHEESE,MEXICAN,QUESO ANEJO
Serving Size 100 g
Amount Per Serving
Calories 373
% Daily Value *
Total Fat 29.98g 38.0 %
Saturated Fat 19.03g 95.0 %
Cholesterol 105.0mg 35.0 %
Sodium 1131.0mg 49.0 %
Total Carbohydrate 4.63g 2.0 %
Dietary Fiber 0.0g 0.0 %
Sugar 4.63g
Protein 21.44g 43.0 %
Vitamin D 0.5mcg 2.0 %
Calcium 680.0mg 52.0 %
Iron 0.47mg 3.0 %
Potassium 87.0mg 2.0 %
* The % Daily Value (DV) tells you how much a nutrient in a serving of food contribute to a daily diet. 2,000 calories a day is used for general nutrition advice.
Nutrition Facts
CHEESE,MEXICAN,QUESO ASADERO
Serving Size 100 g
Amount Per Serving
Calories 356
% Daily Value *
Total Fat 28.26g 36.0 %
Saturated Fat 17.94g 90.0 %
Cholesterol 105.0mg 35.0 %
Sodium 705.0mg 31.0 %
Total Carbohydrate 2.87g 1.0 %
Dietary Fiber 0.0g 0.0 %
Sugar 2.87g
Protein 22.6g 45.0 %
Vitamin D 0.5mcg 2.0 %
Calcium 661.0mg 51.0 %
Iron 0.51mg 3.0 %
Potassium 86.0mg 2.0 %
* The % Daily Value (DV) tells you how much a nutrient in a serving of food contribute to a daily diet. 2,000 calories a day is used for general nutrition advice.
Nutrition Facts
CHEESE,MOZZARELLA,WHL MILK,LO MOIST
Serving Size 100 g
Amount Per Serving
Calories 318
% Daily Value *
Total Fat 24.64g 32.0 %
Saturated Fat 15.56g 78.0 %
Cholesterol 89.0mg 30.0 %
Sodium 710.0mg 31.0 %
Total Carbohydrate 2.47g 1.0 %
Dietary Fiber 0.0g 0.0 %
Sugar 1.01g
Protein 21.6g 43.0 %
Vitamin D 0.5mcg 2.0 %
Calcium 575.0mg 44.0 %
Iron 0.2mg 1.0 %
Potassium 75.0mg 2.0 %
* The % Daily Value (DV) tells you how much a nutrient in a serving of food contribute to a daily diet. 2,000 calories a day is used for general nutrition advice.
Nutrition Facts
CHEESE,BRICK
Serving Size 100 g
Amount Per Serving
Calories 371
% Daily Value *
Total Fat 29.68g 38.0 %
Saturated Fat 18.76g 94.0 %
Cholesterol 94.0mg 31.0 %
Sodium 560.0mg 24.0 %
Total Carbohydrate 2.79g 1.0 %
Dietary Fiber 0.0g 0.0 %
Sugar 0.51g
Protein 23.24g 46.0 %
Vitamin D 0.5mcg 2.0 %
Calcium 674.0mg 52.0 %
Iron 0.43mg 2.0 %
Potassium 136.0mg 3.0 %
* The % Daily Value (DV) tells you how much a nutrient in a serving of food contribute to a daily diet. 2,000 calories a day is used for general nutrition advice.
Nutrition Facts
CHEESE,MEXICAN,QUESO CHIHUAHUA
Serving Size 100 g
Amount Per Serving
Calories 374
% Daily Value *
Total Fat 29.68g 38.0 %
Saturated Fat 18.84g 94.0 %
Cholesterol 105.0mg 35.0 %
Sodium 617.0mg 27.0 %
Total Carbohydrate 5.56g 2.0 %
Dietary Fiber 0.0g 0.0 %
Sugar 5.56g
Protein 21.56g 43.0 %
Vitamin D 0.5mcg 2.0 %
Calcium 651.0mg 50.0 %
Iron 0.47mg 3.0 %
Potassium 52.0mg 1.0 %
* The % Daily Value (DV) tells you how much a nutrient in a serving of food contribute to a daily diet. 2,000 calories a day is used for general nutrition advice.

2.With Diversity = Yes



V. Conclusion and Future Scope

From the above example, it is evident that when we focus on diversity, because of the lack of knowledge-discovery in the nutrition domain, there is a loss of accuracy on using string matching. However, with diversity = No, this project illustrates that with a little deviation on the nutrient values, we can model a recommender system to find alternate food items with similar nutritional values, that can substitute the food item in question. This system can be employed to find alternatives to food which may cause intolerance, for ex. A lactose intolerant individual can replace cheese with tofu which is similar in nutritional value.

With a lot of research in the field of recommender systems, we can further look to use a more powerful type of recommenders, such as a hybrid recommender system, which would take not just the similarity distance, but multiple parameters into account, thereby increasing diversity while maintaining the accuracy of the system.

VI. References

1. Charu C Aggarwal : Recommender Systems, Springer
2. Liang Zhang , Quanshen Wei, Lei Zhang, Baojiao Wang 1 and Wen-Hsien Ho: Diversity Balancing for Two Stage Collaborative Filtering in Recommender Systems
3. https://data.nal.usda.gov/dataset/composition-foods-raw-processed-prepared-usda-national-nutrient-database-standard-referenc-9 : Dataset
4. https://www.fda.gov/food/new-nutrition-facts-label/daily-value-new-nutrition-and-supplement-facts-labels : Nutrition and Supplement labels
5. https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf :Greg Linden, Brent Smith, and Jeremy York: Amazon.com Recommendations, Item-to-Item Collaborative Filtering
6. http://fsnagle.org/papers/celma2010longtail.pdf: The long-tail of recommender systems
7. http://www.scholarpedia.org/article/K-nearest_neighbor: K-Nearest Neighbors
8. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm : K-Nearest Neighbors