NuTriX: A smart way to recommend your Foods

Dr. Mahfuza Farooque Dr. Suman Saha Shubhang Sharma
Phakphum Artkaew Josephine Prickett

November 27, 2020

Abstract

This project searches for a manner in which to predict the dietary restrictions within a food based off an ingredient list. Intuitively, some dietary restrictions are identifiable without question. For example, cheese will trigger a flag for lactose intolerance. The additional step that this project takes is identifying the probability of other dietary restrictions being triggered as a result of correlation between ingredients. To extend the aforementioned example, our machine might detect that a recipe with cheese might also have high sodium content, triggering high blood pressure as a dietary restriction.

1 Introduction

Food plays an important role in our every life; it is essential for the human body to function. People have different preferable food depending on various conditions such as allergens, or congenital disease. Therefore, the food recommendation system is one of the important research areas. There is adequate research related to food recommendation and machine learning, as there is more public data available than before.

The recent research on food recommendations is various. Mostly, researchers were able to predict food's ingredients by taking a picture. They used various machine learning techniques such as SVM, multiclass- Adaboost, etc. One notable work is, "Food-101 – Mining Discriminative Components with Random Forests", which relies on Random Forests Discriminative Components to classify foods into 101 categories. Researchers put emphasis on their use of super pixels, comparable to mosaic tiles, as a stand in for image patches, as they claim are best for object detection.The RFDC model performed well on their own testing data at 50.76% accuracy, only being outperformed by Convolution Neural Network by about 6%. However, when tested on MIT-Indoor data at 58.36%, it was outperformed by Discriminative Mode Seeking and Intuitionistic Fuzzy Values by over 2-8%.With even more fruitful results, " Automatic Chinese Food Identification and Quantity Estimation" relies heavily on sparse coding and SVM to classify foods into 50 categories. Researchers did not compare their models to others developed in the field, but rather compared different variations of their models against each other. They found that Multiclass Adaboost with Top-5 accuracy proved to be the most accurate, with an accuracy of 90.9%. However, the average accuracy of their work is 68.3%.

In this paper, we target to develop a system that can classify food by its ingredients from pictures and texts. We aim to create two machine learning models; first is picture to text model, second is text to text model. Each model will be tested on a different machine learning model and compared to see which one performs better. The classes of model within this paper include model that involve clustering by vectors, statistical methods, and gradient boosting algorithms.

2 Datasets

Our study relies on the data retrieved from an online machine learning platform, kaggle.com. The website

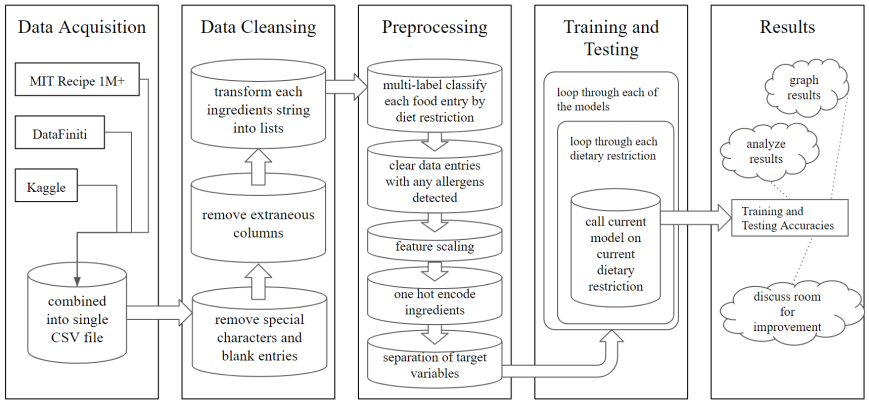

Figure 1: Nutrix System Arcitechture

contains free datasets published by its users, mainly used for data science purposes. We three datasets; the first two were found on kaggle.com, the third dataset was found on eightportions.com. The first dataset, ingredient v1.csv was uploaded by the company Datafiniti under the CC BY-NC-SA 4.0 copyright. Datafiniti compiled 10,000 entries containing attributes such as "brand, name, manufacturer, category, features, and more" (features being the ingredients list). The second dataset was created by Elisa for the purpose of creating a food recommendation system. The dataset was collected from allrecipes.com; it contains about 50,000 recipes, including its name, ingredients, image_url, cooking_direction, nutrition, and id. The third dataset was created by Ryan Lee under the ODC Attribution license. Lee scraped this dataset from various websites; it contains about 125,000 recipes, including name, ingredients, instructions, source, and picture.

In essence, the required data needed in each dataset was an ingredients list for each food entry, which were provided in both of the files collected from Kaggle. As aforementioned, our first dataset contained 10,000 entries and the second one contained 50,000. With that being said, a considerable amount of the entries contained null values for certain values, which was the first obstacle encountered during the project. An even bigger issue that was encountered was that a considerable amount of entries were non-food items (such as Clorox), causing noise in the data.

3 Preprocessing

The preprocessing of data proved to be the most extensive part of the process; the steps taken have been broken down into four parts below:

3.1 Cleaning Data

As a result of the data being sourced through opportunistic methods rather than experimentally, missing and inappropriate data proved to be a cumbersome obstacle. For ingredients v1.csv, there were some ingredients that contain special characters such as î; some of the food was left as none. It might affect the accuracy of the model so first, we removed special characters and none from the dataset. For core-data_recipe.csv, we neglected some columns because the dataset was too big. The columns that were removed were image_url, cooking ingredients, and nutrition.

3.2 Classifying Dietary Restrictions Feature Selections

We created a dictionary of allergens that indicate each dietary restriction and ingredients that contain

them. We aimed to perform a machine learning model on following nine different dietary restrictions; lactose intolerance, celiac, diabetes, high blood pressure, allergens, halal, kosher, vegetarian, and vegan. Then, we performed a string check on each dataset to detect allergens. For example, when we looked at the ingredients of 'Potato Bacon Pizza.', we detected that there was cheese. Thus, we added lactose intolerance into its allergens column. Referencing back to the issue of non-food data entries mentioned in "Overview of Datasets", one more round of data clearing was applied after classifying dietary restrictions. As a fix to the issue, data entries that contained no allergens were removed, since non-food items would not raise a flag on any of the dietary restrictions.

3.3 Feature Scaling

3.4 One Hot Encoding

Since the ingredients column contained entries of lists. The data was therefore three dimensional. To lower the array's dimensionality back down to two, one hot encoding was performed. As a result, each possible ingredient was then listed as its own column and each row displayed a tally of how many times said ingredient shows up in the recipe.

3.5 Separating into Feature Vectors and Target Variables

After One Hot Encoding the ingredients, one more step remained before feeding our data into the models - the separation of variables. The target variable was an array of the different dietary restrictions that was looped through, allowing us to apply the model on each one.

4 Training and Testing Data

One of the challenges faced when searching for correlations of each individual dietary restriction was that the models only identified the correlations for one target variable at a time. However, the process of training and testing the model based off of each restriction followed identical form without loss of generality. Therefore, a for each loop was applied on the set of dietary restrictions, calculating the correlations for each dietary restriction one at a time. The models applied and their hyper parameters are detailed below.

4.1 Linear Support Vector Classifier (SVC)

The Linear SVC model operates by separating mapping data into a hyperplane in which data points can be separated by straight lines. In this research, sklearn's LinearSVC was applied with the default parameters listed within the documentation. The average test accuracy over the ten food restriction classes was 92.41

4.2 Naive Bayes Algorithm

The Naive Bayes Algorithm is rooted in the usage of Bayes' Theorem from statistics. The aim of the theorem is to find the probability than event A will occur on the condition that B has also occurred. In this work, the theorem is used to calculate how probable it is that a dietary restriction will occur given the fact that other restrictions were also flagged. The default settings for the GaussianNB() function from sklearn's naive bayes package was used.

4.3 Catboost

Catboost is a gradient boost algorithm that is primarily used for categorical variables. One hallmark of Catboost is its generation of one-hot encoded variables without the need for external encoding. For the sake of this work, since the all categories were already pre-encoded, the pre-existing one hot encoded categories were fed into this machine. The results of this model averaged at 99.17, proving to be the highest scoring model studied.

4.4 Adaboost (Adaptive Boosting)

In the Adaboost algorithm, focus is aimed towards improving on the gradient boosting algorithm itself by assigning weights to each classifier based on its tested accuracy. In this work, the Adaboost algorithm was used with 100 estimators and a random state of zero. The model took the longest to run, yet yielded the lowest results - averaging at 45.65 percent. Additionally, the results lacked precison, facing lows of 4.6 percent accuracy for Kosher and maxing at 93.6 percent for Halah.

4.5 Test Results

	Gaussian Naive	LinearSVC	CatBoost	Adaboost
Diabetes	64.36	92.51	99.87	27.47
High Blood Pressure	69.95	93.6	99.74	52
Halah	75.22	97.88	99.95	93.6
Celiac	96.85	99.48	99.97	5.91
Allergens	60.26	94.14	99.22	74.96
Lactose Intolerance	55.37	88.48	99.58	33.12
Vegan	39.21	72.88	95.21	41.1
Vegetarian	66.66	92.85	99.04	78.13
Kosher	95.31	99.88	99.99	4.6
Average	69.24333333	92.41111111	99.17444444	45.65444444

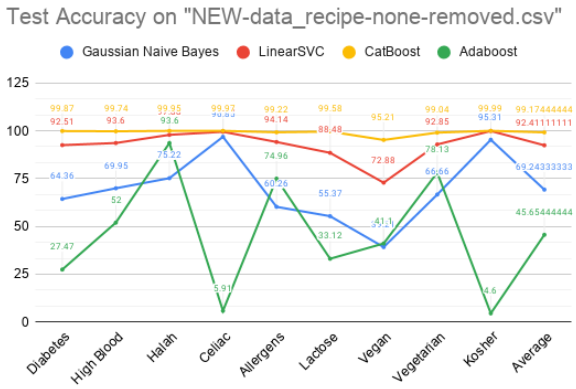

Figure 3: Test Results on Each Diet Restriction

On the other side of the spectrum, Adaboost yielded the lowest results. Not only did the results prove to be inaccurate, they also proved to be extremely unprecise, opposite of Catboost.

These results spark much interest, as both Catboost and Adaboost share many similarities, both being gradient boosting algorithms, yet seemed to have polar opposite reactions to this dataset.

5 Postprocessing and Visualization

After each model was run, the testing accuracy was recorded into Google Sheets. The averages for each model throughout each food group was calculated and the results were plotted.

6 Results

As seen in the above figures, Catboost proved to perform superior to all other models on the dataset. Not only did it score an average test result that was higher than all other, it scored higher accuracy than any other model for each dietary restriction as well. As mentioned Section 4.3, Catboost is design specifically for categorical data. Hence it's performance.

7 Conclusion

In conclusion, further research could include the fine tuning of hyper parameters within the models being used.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.