Hoos Fit*: A Machine Learning-Based Personal Dietitian - Checkpoint

Vijay Lingesh
Dept. of Computer Science
Univ. of Virginia
Charlottesville, US
vb7mz@virginia.edu

Goutham Patley
Dept. of Computer Science
Univ. of Virginia
Charlottesville, US
gp4em@virginia.edu

Leonard Ramsey
Dept. of Computer Science
Univ. of Virginia
Charlottesville, US
lr3hj@virginia.edu

***Abstract*—'Hoos Fit' is an application that will assist UVA students in reaching dietary goals. It will use the information available from product nutrition labels to help users make informed dietary decisions through a recommendation system.**

***Index Terms*—nutrition, fitness, nutrition label, OCR, image processing, machine learning, classification**

I. Motivation

Nutrition is important at every phase of life. It helps us stay stronger and feel better. While regular exercise is important, maintaining a balanced diet is essential for staying healthy. This is emphasized more for college students, who are sometimes too busy studying or completing other activities to research a good diet. Staying fit is more about eating smart than working hard. Hence, we present a solution which will help UVA students make smart decisions from the information that is readily available but hard to interpret.

II. Method

We divided our approach in completing the project into three phases: designing a user interface, building a classification algorithm to score food and beverages, and completing image processing on nutrition labels. Up to this point, we have focused on completing the first two phases.

The first phase of our project consists of designing a Django web application. The purpose of this web application is to accept nutrition label images from a user and respond with a score retrieved from the output of our machine learning algorithm.

The second, largest and most important phase of our project consists of completing the end-to-end machine learning project for scoring nutrition label information. First, we need to scrape our dataset based on data available in Open Food Facts as we do not have a centralized dataset readily available to download [1]. This data consists of features like 'Carbohydrate, Dietary fiber, Energy, Fat, Proteins, Salt, Sugars, Nutrition Score,' etc. We will be focusing on the Nutrition Score as the predicted label from our hypothesis. The Nutrition Score is a categorical rank provided to a food product based on the nutritional composition. The Nutrition score ranges from A to E, although in our pre-processing of the data, we will convert these ranks to 5 to 1 (where A maps to 5 and E maps to 1).

Since our predicted value is categorical, we cast our problem as a supervised learning classification problem. In order to model the classifier from the nutrition features, we will run them through various classification algorithms.

Our first ML phase would be to clean the data for training. Different products have their own set of nutritional contents which may be absent in another food product. Thus, the data needs to be imputed and truncated where appropriated In order to highlight the relationships between features, we must also scale the data via standardization.

Once our data is cleaned, we must then proceed by selecting and training various models from various classification algorithms available. Our intuition tells that there would be a non-polynomial co-relation between the Nutrition Score and the nutritional components, so we will start by training and test a logistic regression model as the baseline model for training. After getting the accuracy, precision, and recall of logistic regression, we will train the dataset with various other models such as Linear SVM, RBF SVM, Decision Tree and Random Forest. Finally, we will try to use various ensemble learning models and compare to see which model is the best.

III. Preliminary Experiments

Our preliminary steps focused on completing the first two phases described in Section II.

For the first phase, the UI of the web application is stable and images can successfully be uploaded. We also have the infrastructure in place to send an uploaded image to an AWS EC2 instance to be processed in the machine learning algorithm, and to receive a response.

For the second phase, the data has been scraped, read in, pre-processed, and we have tested a numerous number of classifiers on our data so far. We designed a Python web scraping program using the Scrapy framework to sweep through each of the Open Food Facts product pages and extract the nutrition label images and tables that were provided. We also had to clean the data we scraped by first ensuring any possible nutrient component was added as a feature to our dataset. If the nutrient is not a part of the product, we have the value as 0. We experimented with dropping rows with a certain number of missing nutrients to ease the pre-processing and retain the integrity of the remaining data in training the

models. We also dropped features that were crucial to scraping the data but not for the machine learning algorithm, such as the url path of the product on the website and the file name associated with the product on the website. We also had to clean certain cell entries that came along with any html special character tags such as &lt or &gt in the dataset. For example, for certain products, the sodium content was mentioned as < 1.8 mg . This does not give us specific data and is not really useful when useful when training our model. Since such trends with &lt or &gt occurred rarely, we safely placed them to be zero. We also had to remove any entry that did not have a Nutrition Score as well as without the label, those entries are of no use.

Once we had our cleaned data, we split the data into training and testing sets with X being all the nutrients and Y being the Nutrition Score as the label. We found the correlations of all nutrients with the Nutrition Score. The strongest correlations are reflected in Table 1.

Nutrients and Food Components	Correlations
Biotin	0.998777
Fruits, vegetables and nuts (minimum)	0.705018
Pantothenic acid / Pantothenate (Vitamin B5)	0.550535
Vitamin E	0.519808
Selenium	0.427489
Monounsaturated fat	-0.280155
Energy from fat	-0.43868

Table 1: Features with strong correlations to Nutrition Score

We dropped the least correlated values and started our training with some classification algorithms that come as built in libraries. We started with modeling the dataset in a logistic regression function, and tested decision trees, random forests, linear and RBF SVM, and several ensemble learning models. For each model, we calculated the precision, recall and accuracy scores. A few of the metrics are reflected in Table 2.

Classifier	Accuracy	Precision	Recall
Logistic Regress	0.8364	0.8331	0.8364
Linear SVC	0.8364	0.8445	0.8364
RBF SVC	0.8553	0.8521	0.8553

Table 2: Accuracy, precision, and recall for several classifiers

We wanted to apply better algorithms as we can't know for sure if logistic regression is the right way to go without comparing the scores with other models. So, having the logistic regression data as base line we ran the following algorithms and calculated their Precision, Recall and Accuracy scores as well.

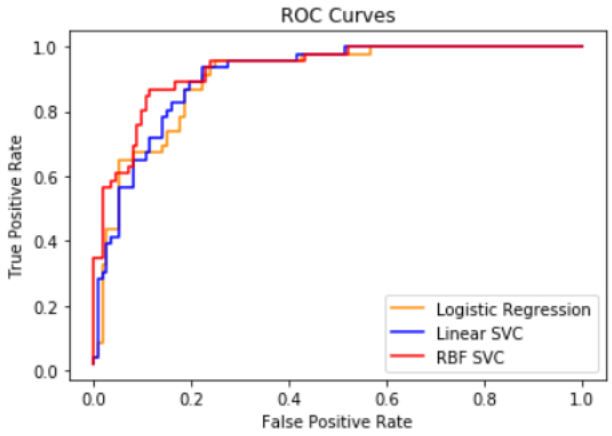

Figure 1: ROC Curves for classifiers tested from Table 2

IV. Next steps

The next steps involve fine-tuning our models, testing more models, completing the image processing of the nutrition labels, and connecting all of our modules together to build a complete system.

We want to make better models until we get the best precision. We will be proceeding with fine-tuning Decision Trees models, Random Forest models and ensemble learning models to see if that will be providing the best possible precision we can get.

As we finish testing and selecting the model to use in our final system, we plan on starting the image processing component of the system. This image processing is essential as the program will have to be able to read all the characters in the image and extract the text from it in order to feed the machine learning algorithm. We will try to use machine learning to do OCR on the input image to retrieve the nutrition components and values. If time does not permit us completing this step from scratch, we will be using third party tools, such as Tesseract, to read the characters from the nutrition label image. After the OCR processing, we use the characters read from the image as X test and fit this into the model that we refined and then predict the Nutrition Score of the new food product that was passed as input to the web application. Based on the score, the user will receive the nutrition score with dietary recommendations about the food product and the user can choose to consume the product or not according based on fitness goals and needs.

If time permits, we will also add a user profile component to the web interface, so the user can have a more personalized experience complete with recommendations related to pre-established preferences.

References

[1] Open Food Facts - United States. (n.d.). Retrieved from https://us.openfoodfacts.org/.