Capstone update: Predicting recipe sentiment from nutrient composition

Mike Touse

February 18, 2017

Following initial data capture and exploration, the topic was modified to create a model that predicts recipe ratings based solely on nutrient content. This approach yields a model that is much more general that can be extended or mapped to recipes as well as prepared or packaged foods.

1 Initial data source investigation

The USDA National Nutrient Database provides extensive lists of foods and nutrient data through 12 ASCII files available for download and use. By querying individual foods across multiple files using food description and nutrient identification numbers, one can extract desired features (such as weight or specific ingredients / nutrients). Database files were downloaded, parsed into pandas DataFrames, and explored for utility.

In order to build a classification model to support selection of individual products or recipes, however, a collection of product ratings are needed to train the model. The only rating sources discovered were from recipe websites which, unfortunately, did not couple ratings, ingredients, nutrients, and weight in a common dataset suitable for training the desired model. Sites investigated (with appropriate api's and usage terms) included food2fork, edamam, and Yummly.

2 Updated Client and Problem definition

The purpose of this analysis is to develop a predictive model that can be used to identify the popularity of foods based on their nutritional content or other features. The development of such a model is broadly applicable throughout the food industry, but primarily geared towards clients who directly provide recipe content such as recipe websites.

With nearly 100M monthly site visits to a top recipe website, improving site membership and click-through-rate (CTR)can generate significant revenue for minimal investment. Predicting recipe performance even before receiving user ratings would allow the company to highlight new recipes with reasonable confidence that the recipe would improve overall site utilization and yield higher CTRs.

3 Data collection

Ultimately, Yummly.com provided the most appropriate data to capture recipe information paired with rating data. Combining data from multiple sites might yield more complete information (including weight), but will require extensive manual querying to adapt between different site identification schemes.

Yummly.com uses two different API's to provide responses (in JSON format). The Search Recipe API returns a list of recipes that meet the search criteria and provide some attributes of the recipe. The Get Recipe API uses a single ID provided from the Search Recipe response to return the detailed nutritional data and recipe text details.

Yummly Scripts were developed to handle api requests within the education license term limits through the following process:

- Query subsets of recipes using individual ingredients (such as beef, chicken, tomatoes, etc). Responses included ratings and recipe id's.
- Concatenate subsets into single list of recipes.
- Query individual recipe id's from concatenated list to capture complete recipe information.
- Parse necessary fields into individual components (namely, the quantity of individual nutrients).

Following an initial collection of the recipe list, it became apparent that the recipe ratings were highly unbalanced (with nearly all recipes rated 4/5 and a smaller percentage rated 5/5). Further investigation suggested that selecting recipes towards the end of the api response tended towards lower ratings and the bulk collection scripts were adapted to balance the rating distribution of the dataset to allow improved model training.

Even after adding lower-ranked data to the set, most of the recipes were rated either 3/5 or 4/5. This would likely give poor results if attempting to predict actual recipe ratings, so the goal was shifted to a binary classification of either 'Favorable' (4/5 or 5/5) or 'Unfavorable' (0/5 - 3/5).

4 Data Structure

Once the recipe list was compiled using the Search Recipe API, the individual recipes were downloaded and the json responses were pulled into a pandas dataframe. The nutrient information was contained in a list within a single column and required further parsing. Ultimately, a dataframe was compiled that contained the recipe ID, rating, number of servings, preparation time, and individual nutrients, each as a separate feature (column). The recipe response only populates fields for the nutrients listed for the individual recipe, so the dataframe was constructed to include every nutrient listed, with many of the fields populated with null ('NaN') values. Nutrient values were normalized to a 'per serving' value using the 'number of servings' field, and null elements were filled

with a value of '0'. Finally, recipe ratings (0-5) were mapped to 'high' or 'low' ratings as previously discussed.

5 Data Exploration

Initial exploratory data analysis focused on examining which nutrients were reported in the largest percentage of recipes and the distribution of ratings across each feature in order to identify potential predictive features. Results from this analysis revealed very little correlation between any particular nutrient and the rating/class label.

6 Analysis plan

In order to identify weights of particular features that can be used to classify individual recipes, the first step will be to attempt to fit a logistic regression to the nutrient data. Because of the large number of features (¿100), L1 regularization is expected to yield more useful results by driving some portion of individual feature weights to zero. Depending on results of the logistic regression classifier model, the next step will be to attempt to classify recipes using a random forest.