Capstone 1 Project
Milestone Report

by Anna Kantur

date: 01/20/2020

Problem Statement

We want to analyze attributes of multiple recipes and the corresponding recipe reviews. The result of our analysis would be a summary of what successful recipes have in common: e.g. is it nutrition value, number of steps required, time required, a specific ingredient, etc.

Our analysis will have a broad range of clients from a grocery store owner who decides what products to put on display for advertisement to professional and amateur cooks who are trying to create a menu with the best recipes.

The deliverables for our project will be a Juniper notebook with code and a paper summarizing our approach and findings. We will also have slides for presentation purposes.

Dataset

We will use the database of recipes RAW_recipes.csv and the database of recipe reviews RAW_interactions.csv from Kaggle:

https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions

This dataset consists of 180K+ recipes and 700K+ recipe reviews covering 18 years of user interactions and uploads on Food.com (formerly GeniusKitchen) The information presented in the dataset includes recipe attributes and recipe ratings from multiple reviews.

Data Wrangling

We uploaded 'RAW_recipes.csv' and 'RAW_interactions.csv' files from into Pandas data frames. For the purposes of our analysis we decided to keep only the recipes with more than 10 reviews. The original dataset had 210,244 recipes, but only 18,762 with more than 10 reviews. We created recipes_df and reviews_df in pandas with recipes that met this criteria.

The description column of recipes_df had 406 empty values. We decided to keep recipes with missing descriptions in the data.

Recipe Tags

We investigated the ‘tags’ column of the recipes dataframe. The output looked like a list, but the type of the data was a string. There were 495 unique tags from all the recipes.

We focused on creating a tags_matrix data frame with recipe ids in rows, all individual tags as columns, and values being 1 if the ingredient is present in the recipe and 0 if not. To do this, we performed the following steps:

1) Added a column tags2 to the recipes data frame where the values from the tags column were converted to a list

2) Added a column tags_check to the recipes data frame with a list of unique tags from all the recipes
3) Created a function that run through all the values in the tags_check column and replaced the tag value with 1 if it is present in the recipe and with 0 if it is not present. Tested that the function is working on one recipe id.
4) Applied the tags_check function to the column tags_check using list comprehension to get to the individual tags level
5) Created the tags_matrix data frame with rows as recipe ids, columns as the individual recipe tags, and values as 1-s and 0-s for a particular recipe id-tag match from the tags_check column. We doublechecked that the tags_matrix has correct values by comparing the output of the tags_matrix for one recipe id to the sum of tag values from the tags_check column of the recipes dataframe for the same recipe id

Recipe Ingredients

We investigated the ingredients column of the recipes dataframe. Similar to the 'tags' column, the data output of the 'ingredients' column was a string. We created a list with 8,091 unique ingredients from all the recipes. We found that this list has similar ingredients with different names. We decided not to modify the ingredients column itself and address the issue further by data wrangling the ingredients matrix.

We narrowed down the list of the ingredients to only the ones that are present in more than a 100 recipes. There were 245 of such ingredients.

The ingredients matrix was created similar to the tags_matrix. We then cleaned up the ingredients_matrix by getting rid of:

1) Ingredients with names in multiple (e.g. 'eggs' vs 'egg')
2) ingredient with the same first word in the name (e.g. 'garlic powder' vs 'garlic')
3) ingredient with the same second word in the name (e.g. 'fresh cilantro' vs 'cilantro')
4) ingredient with the same third word in the name (e.g. 'freshly ground pepper' vs 'pepper')

We ensured that the values from the removed columns were correctly added to the relevant ingredient column in the ingredients_matrix. E.g. if 'garlic powder' column had 1 and 'garlic' column had 0 for a particular recipe, then the 'garlic' column would have 1. The resulting ingredients_matrix had 149 columns.

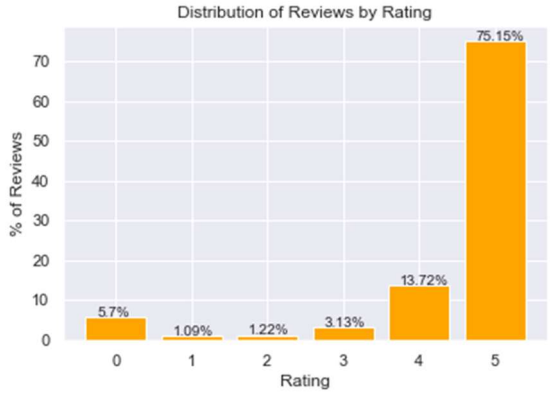
Files for further use in Capstone 1

As the result of the data wrangling we prepared several files for further analysis:

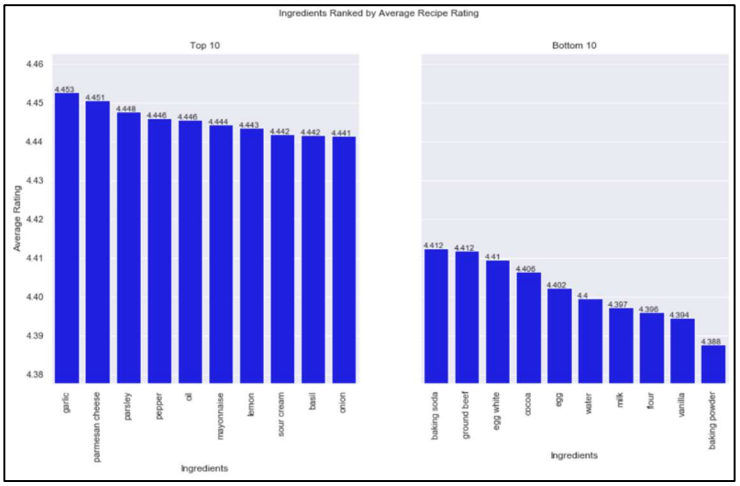
- Recipes_df
- Reviews_df
- Tags_matrix
- Ingredients_matrix

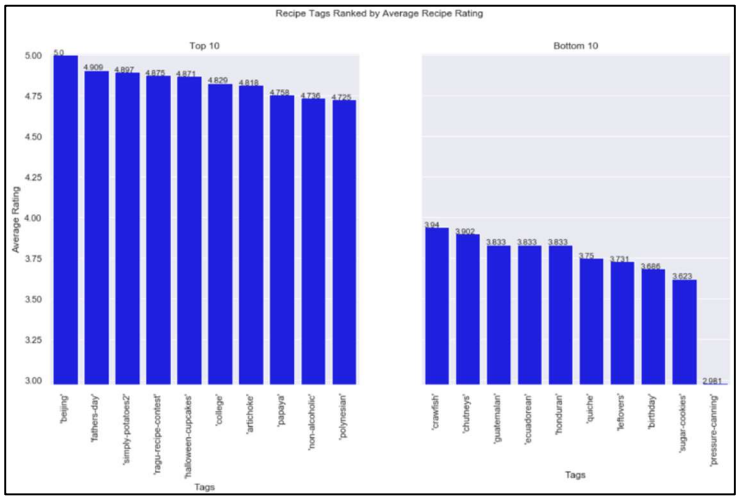
Exploratory Data Analysis

Distribution of ratings for all the recipes shows that 80%+ of reviews are 4 and 5 stars.

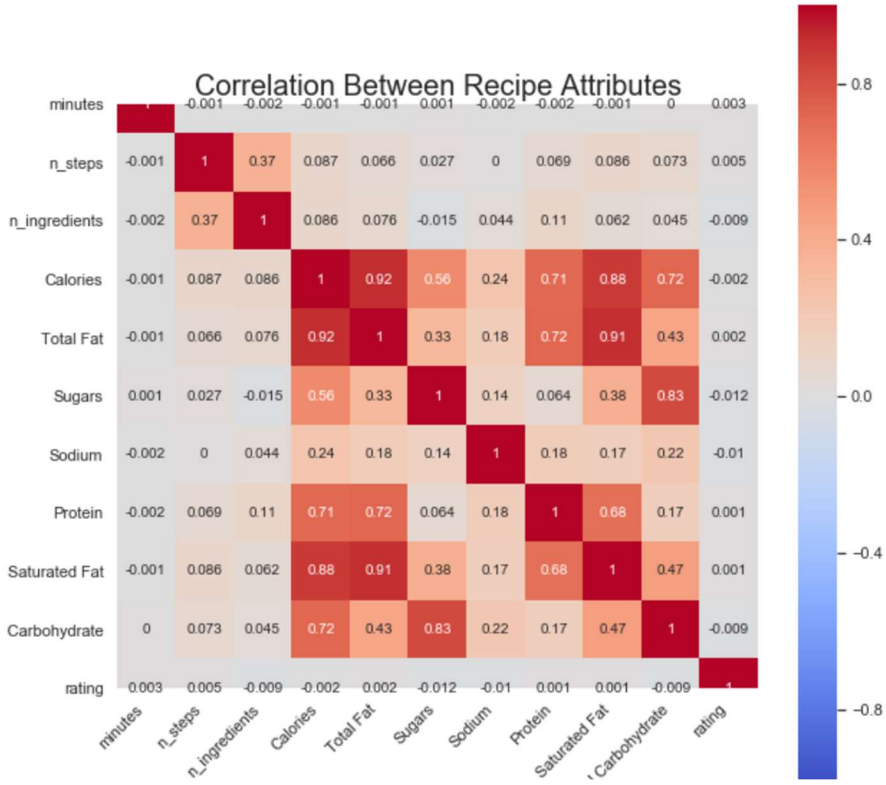


We identified the best and the worst ingredients and tags by recipe rating.





We also showed the nutritional values correlation matrix. There is a strong correlation between calories and the total fats (specifically saturated fats), protein and carbohydrates contents. However, there is no strong correlation between the recipe rating and any of the mentioned recipe attributes.



We further investigated the data in EDA and have the following findings:

1. The best ingredient is garlic. It has the average recipe ranking of 4.453
2. The worst ingredient is baking powder. It has the average recipe ranking of 4.388
3. The best tag is 'beijing'. It has the average recipe ranking of 5.000
4. The worst tag is 'pressure-canning'. It has the average recipe ranking of 2.981
5. Successful recipes on average take longer to make and include 10 or more steps
6. In terms of nutritional value, successful recipes have more total fats, sugars and carbohydrates, but less sodium and saturated fats, and slightly less protein.

We performed two sample t-tests to confirm our EDA findings:

1) The two independent samples were the recipes with a specific attribute and the recipes without it

2) We tested the null hypothesis H_0 that the mean recipe rating for the two samples is identical

2) The alternative hypothesis H_a was that the rating means are different (e.g. the mean recipe rating is affected by a specific recipe attribute).

3) If the t-test results were statistically significant (e.g. p-value > α, α = 0.05), then we rejected the H_0 and accepted the H_a.

As the result of the t-tests:

1. **Garlic** The test had p-value of 9.475783077096418e-18 which is very small, so we can reject the H_0 and confirm that garlic in recipes contribute to a higher recipe rating

2. **Baking Powder** The test had p-value of 9.475783077096418e-18 which is very small, so we can reject the H_0 and confirm that baking powder in recipes contribute to a lower recipe rating

3. 'Beijing' recipe tag The test had p-value of 0.13 > aa = 0.05, so we can reject the H_a. This means that 'beijing' tag positive effect on the recipe rating is statistically not significant.

4. **'pressure-canning' recipe tag** The test had p-value of 2.792736995754129e-10 which is very small, so we can reject the H_o. This means that 'pressure-canning' tag negative effect on the recipe rating is statistically significant

5. **Saturated Fats Content** was the only other attribute that affects the recipe rating in statistically significant way (the test p-value was 0.03). All the other recipe attributes did not significantly affect the recipe rating.