Datasets cleaning, merging and analysis description

The collected datasets were scraped or downloaded directly from the respective source. This means that a lot of junk like useless information, html code, unicode/ascii tags, incomplete data were introduced.

In the Whatscooking dataset every "\u00a0" and "\n" in the description were replaced with white spaces or empty spaces and then merged into a single string representing the entire description for the given recipe. The servings and calories values were transformed or at least treated as integers, every other nutritional value, instead, was saved as float. In whatscooking also an estimated cost is given for each recipe as column. This cost was normalized into a value from 0 to 100. There is a high chance that this field will be removed entirely during the delivery of the project since it is pretty pointless. The directions of a recipe is slightly messed up by unicode special characters, which needed to be removed. Therefore, every "\u00a0" was replaced by a white space and every "\n" was used to split a direction from the following one, generating a complete array of strings, each containing a direction for the given recipe. It is explained later how ingredients were processed since it is a main part of the cleaning and should be compared with the other datasets aswell.

The thekitchn dataset, similarly to whatscooking, is a obtained by scraping a website. Therefore, columns like calories needed to be saved as integers and other nutritional values as floats. The description was scraped as a list of strings, so every string got merged into a single one with "<br>"s instead of "\n"s since rapidminer doesn't like unicode characters. In the ingredients list, some "null" entries were detected. So, they were removed and if no ingredient remained, the entire recipe got deleted. Going even further, some recipes had only a title, without any other information: these were removed aswell.

The epicurious dataset is the downloaded one from kaggle. The entries were already cleaned, besides of unicode characters in the description and some recipes containing only a title and nothing else. So, basically, the cleaning and data adjustments were similar to the previous ones.

The ingredients are stored in every dataset as an array of "quantity in number", "unit" and "ingredient". Unfortunately in a couple of datasets this information is not well-formatted: quantity and unit could be in the same value or, even worse, everything is in the same value. In addition, keeping the ingredients in an array inside the table of recipes is inconvenient for different reasons. The idea is to refine the ingredients in a more logic way and to save these in a separated table, with a link to the original recipe. It can be easily seen as a one-to-many table with an id refering to a recipe and 3 values refering to the quantity of an ingredient (with its unit). With some regular expressions all these ingredients information were properly formatted to fit into the desired structure "quantity", "unit", "ingredient". It could happen that ingredients suggests to use "some of this ingredient OR some of that". For simplicity only the first choice was taken for every of these situations. An example could make these example clearer: if a recipe contains id, description, some nutritional information, a number of servings and an array of ingredients in the form of "200 g of pasta", in the new generated ingredients table there are several rows for every ingredient of this recipe, one of these is id and "200", "g," "pasta".

Cleaning happened for the most part during the importing from the local json file to rapidminer, so basically in the importer script in python. An id was created for each entry aswell. To recognize from which source each recipe come, the id is "ep"+seq\_no for the epicurious dataset, "wc"+seq\_no for whatscooking and "tk"+seq\_no for thekitchn.

Moving to the analysis of data it is important to talk about the generation of a label containing the "type of cuisine" of a recipe. Since this relatively type of information is missing in every dataset, a machine learning algorithm is used to predict this information. In this chapter the data analysis will be described, while the process and the training set will be shown in the data integration chapter, which is more appropriate for this kind of manipulation.

After the described cleaning, in the whatscooking dataset 1526 recipes are stored. Only 3% of these are without a description, 308 rows don’t have an estimated cost (even though this information isn’t really relevant in this project), around 100 recipes don’t have complete information about nutritional specs (like calories, proteins, …) and only 2 rows are without directions on how to cook the recipe.

In the Thekitchn dataset there are a lot of missing values. In the 948 recipes, 61% do not have a description, no one has an estimated cost and 41 rows have missing servings or nutritional information. Only 1 recipe has no directions.

Epicurious is the dataset with 20054 recipes. 33% of them have missing description, servings and estimated cost are not present at all, about 4000 rows have missing values in the nutritional information. Only 11 recipes don’t have directions.