# Dataset extraction, generation and description

For the extraction of information, 2 websites and 1 existing dataset were used. The hardest part of this task was to find existing datasets with enough data and information needed for the described scenario. A lot of them were small despite the fact of having a lot of information. The bigger ones usually had less features but, at least, a lot of rows. By searching websites, instead, there were even more constraints. A website needs to be as static as possible since web scraping through dynamic websites is a lot more sophisticated, especially figuring out the URLs of each recipe. In addition, it is easier if for each recipe link there is a simple XPath query to get a specific information from it. During the research of cooking websites, some incredibly weird html structures were discarded because XPathing or analysing URLs were quite difficult.

The existing dataset is taken from Kaggle, a data repository for data scientists which contains a lot of different datasets. This dataset is called epicurious, since it is scraped from a well-known cooking website “epicurious.com”. There are 20’000 recipes containing various information like a title, some calories/protein/fat information, some guidelines on how to cook the recipe and a list of ingredients. This dataset was simply and directly downloaded as json file. Then, the dataset was analyzed and well-formatted with a simple json formatter for readability.

The other 2 datasets are taken directly from websites through a procedure of web scraping. A python script scrapes through a website by collecting the links of every recipe, then it analyzes asynchronously every page through XPath queries to collect various information about each recipe: title, directions, ingredients and so on. The 2 scraped websites are "whatscooking.fns.usda.gov" and “thekitchn.com”. For each of the 2 websites the python scripts are very similar, with some adjustments to collect every link of recipe and some adjustments about the XPath queries. In both cases a json dataset is created which is stored locally. These 2 datasets have 1226 and 999 recipes

A python script is used to import a json dataset into Rapidminer. This is the most suitable procedure found out, since Rapidminer is not the user-friendliest application in importing non-csv/xml files. By translating a json file into a pandas dataframe, Rapidminer can easily read the resulting dataset.

Some cleaning and refinement were already done during the extraction procedure, but this is explained later in the next chapter. The resulting datasets contains the following data:

Epicurious: title, description, a list of ingredients with quantities (not really well-formatted), calories, protein, sodium, a user rating from 1 to 5.

Thekitchn: title, description, servings, calories, labels, fat, saturated, carbs, fibers, sugar, protein, sodium, estimated cost, directions, ingredients (in a well-formatted way with number, quantity unit, ingredient)

Whatscooking: title, description, directions, servings, estimated cost, calories, fat, protein, sodium, ingredients (divided in quantity and ingredient)

Transforming the ingredients into a well-formatted table with the right quantities, quantity unit and type of ingredient is shown in the next chapter since this kind of data refinement was done after the extraction of datasets.

Links:

Epicurious source: path

Epicurious importer: path

Whatscooking source: path

Whatscooking scraper: path

Whatscooking importer: path

Thekitchn source: path

Thekitchn scraper: path

Thekitchn importer: path