Problem formulation

For network analysis part we want to identify the users with the highest engagement within specific recipe communities.

The purpose of the NLP part is to find the patterns in the ingredients of the recipes to find cluster similarities between them. We want to predict based on the rating of the ingredients list which are the ingredients that people prefer most.

Description of the data

The dataset consists of over 180.000 recipes and more than 700.000 recipe reviews which span over 18 years of users interactions posted on Food.com. (Bodhisattwa Prasad Majumder, 2019)

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

The first dataset is about the users' interaction with their reviews on certain recipe_id. The second one is the table of all recipes available and their characteristics such as recipes' name, how long each recipe takes (minutes), the user who posted the recipe (contributors_id), ingredients and so on.

Analysis structure

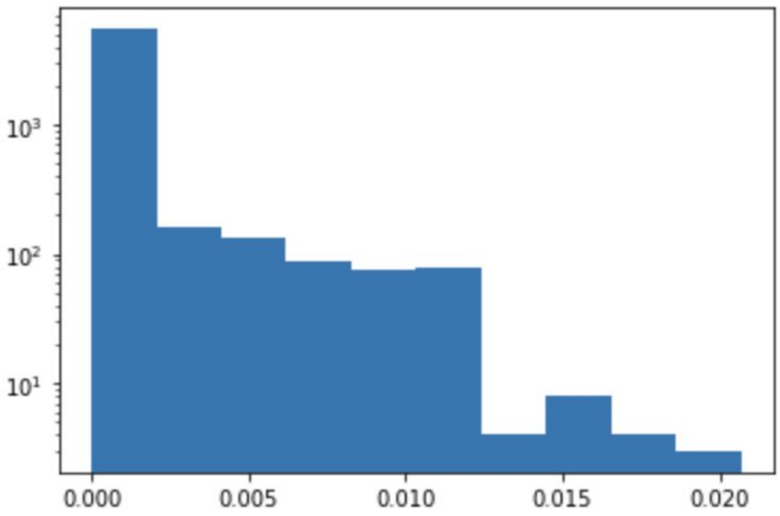
1. Network analysis

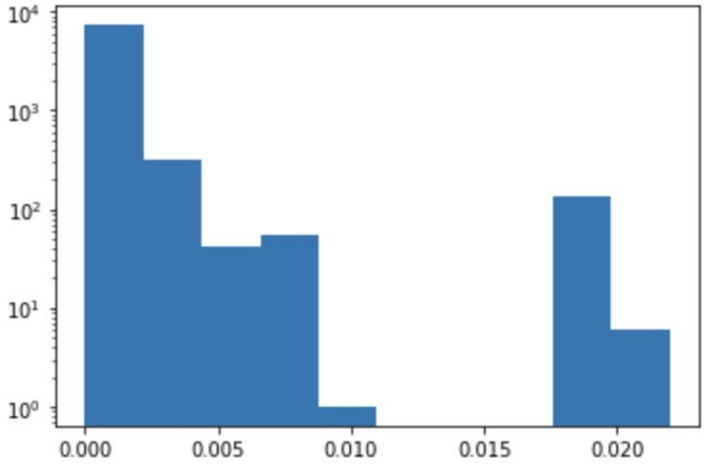
Pre-processing steps. Before starting the network analysis part there where a few steps we had to take. Due to the amount of information we get from the datasets we decided that it would be enough to analyse the data from June 2018 until present day. We have tried to include more data, however that made it difficult to run some of the codes as the Collaboratory could not process it efficiently.

Analysis. Here we created the network with recipes and users as the elements of the graph. To be more specific, we created a bipartite graph between users and recipes. Then we created the weighted projected graph with users nodes partition and recipes one then calculated and analysed the centrality measures on such graphs.

After that, we implement community detection based on how active a user is, in other words, how frequently they make a review on a certain recipe. We did try to plot some graph about centrality measurements. However, due to the huge data, it took us too much time to execute and ended up with a completely black image. Therefore, we only presented the graph with degree centrality for 2 nodes sets which are user_nodes and recipe_nodes as below.

The x-axis is the degree centrality scores and the y-axis is the number of vertices.





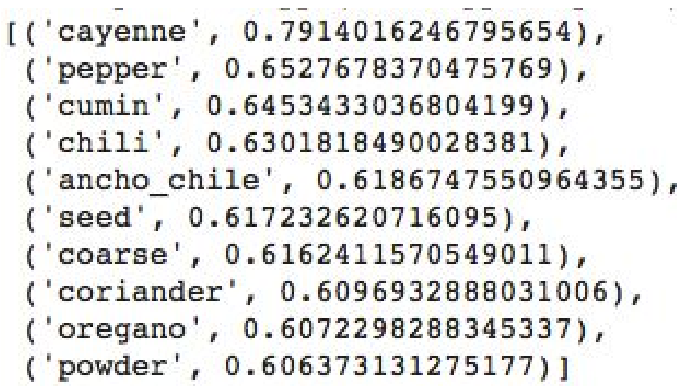
2. Natural Language processing

Pre-processing steps.

In the preprocessing part there was not that much to do, since the ingredients were written in one separated column, which saved us a lot of time. If the ingredients would we written as a part of the whole text, we would need to do the entity recognition and we would need to look for NOUNS and somehow just extract the nouns that are relevant as ingredients. But because they were written in one column, for this part we only needed to merge two different data frames on recipe_id to get the rating of the recipe in the one row as the ingredients. We did some punctuation removing and also out of the one million rows we chose 10,000 from each rating. We continue for the main part from here.

NLP necessary steps

We did some Bag of Words using vectorizer and making our corpus (the whole text) out of this. It is in order to make it easier for a machine to understand the text. Some more NLP steps like TF-IDF, LSA and LDA needed to be done in order to get some interpretable results. LDA splitted the ingredients into the topics based on their similarities and we can see in some of them some pretty good connections. For example we can clearly see the recipes for baking cakes, making lemonades or seasoning meat. The cake baking recipes is the biggest cluster with most of the tokens, therefore we can assume that the biggest group of recipes from this page was for cakes (yummy :P). We also train our own model - model paprika, to present us with the most similar words to other words. We chose paprika and the results are pretty neat, the number next to the word represents on how many % the word is similar.



```
[('cayenne', 0.7914016246795654),
 ('pepper', 0.6527678370475769),
 ('cumin', 0.6453433036804199),
 ('chili', 0.6301818490028381),
 ('ancho_chile', 0.6186747550964355),
 ('seed', 0.617232620716095),
 ('coarse', 0.6162411570549011),
 ('coriander', 0.6096932888031006),
 ('oregano', 0.6072298288345337),
 ('powder', 0.606373131275177)]
```

Unsupervised Machine Learning

For the unsupervised ML we did clustering with KMeans clusters, which takes the closest “neighbours” of the word and tries to put those into certain groups by the similarities. We found 9 clusters and then we printed the recipe's ingredients that contained those ingredients, that appeared in one of the 9 clusters. So we could see the recipes, that contained the most used ingredients from all of them.

Supervised Machine Learning

In this part we had the biggest upset. We ran two different models - Logistic Regression and MultinomialNB to try to predict a rating based on the ingredients. Both of them gave us score only about 20%. This is very low :-/

There might be several reasons for it:

1st - The problem might lie in even distribution of the recipes. Remember in the beginning we had 800,000 5stars recipes, but only 12,000 1star recipes. We wanted to “level” the field, so we took only 10,000 recipes of 5stars, 10,000 of 4stars, 10,000 of 3stars etc. Maybe the algorithm would need to learn from all the 800,000 5stars in order to predict it better - on the other hand, it takes now 16min to load the notebook with 60,000 recipes. Imagine would it would do with 1,1million recipes…

2nd - The problem is that the spectrum is too broad. The difference between 5star and 4star recipe doesn't have to be so significant. As well as 1star and 2star recipe. That's why the algorithm might have it very difficult to predict the difference of these recipes. Maybe if we only take the 5star and 1star recipes, the different attributes of them would be so significant, the model could much easier predict, whether this is a bad or good recipe. We think this is how we should have done it and for a next project, we would definitely look at the two opposite - tasty and not tasty recipes, and make much better prediction for future cooks :-)