Data Clustering and Similarity Matrix Construction

I decided to use my own set of reviews as it was suggested in the Task 2 instructions. First, I ran the Yelp processing script (p27_processYelpResaurants.py) numerous times in order to get a decent number of cuisine category files to choose from. I obtained reviews for 164 cuisines. Then, I selected cuisines for my analysis. The selection criteria included: 1) the number of reviews - the more reviews the better (there is an increased chance of random illogical illogical results if you operate on a small dataset), 2) the interestingness of a cuisine for comparison purposes, for example I tried to include intuitively similar cuisines in order to see if my text mining methods will be able to reveal whether they are similar indeed.

For this purpose I selected Mediterranean / Greek / Middle Eastern / Turkish cuisines, Mexican / Tex-Mex / Latin American / Venezuelan cuisines, Fast Food / Food Court / Food Stands / Hot Dogs cuisines, Chinese / Cantonese / Korean cuisines, Indian / Pakistani cuisines, Vegetarian / Vegan cuisines, and the Russian and Ukrainian cuisines (although the number of reviews was not very big for these two - but this is the only case when I included cuisines with a small number of reviews just out of curiosity). Of course, some of the above groups of cuisines may not be too similar only because of the geographical proximity, but I decided to test this assumption and make my study more interesting. When my code was reading the cuisine cuisine files, it concatenated all the reviews for a given cuisine.

I used the ggplot2 along with reshape2 libraries of the R language and, if needed, matplotlib for the purpose of visualization. For R, I saved my similarity matrices in a file to be used later for plotting. One can also call R directly from Python using the rpy2 module [2].

Task 2.1

I consider this step preliminary, as more parameter tuning will be done in Task 2.2. Here are the results I obtained. For Figure 1, I used the TfidfVectorizer without IDF (but with stopword removal); Figure 2 - with IDF; and the gensim module's LDA model was used for Figure 3. The TfidfVectorizer used by the LDA model had the same parameters as in the IDF case.

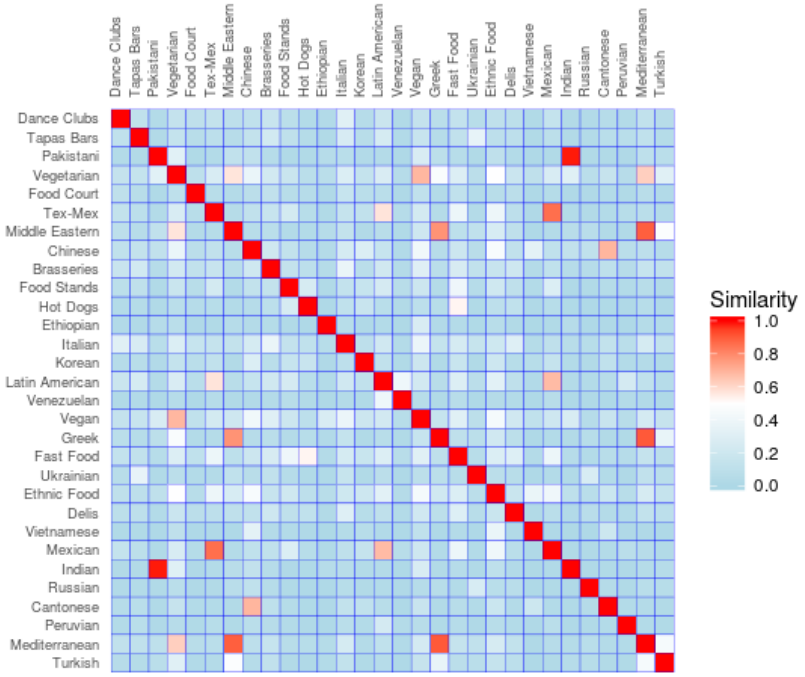


Figure 1. Similarity Matrix from noIDF

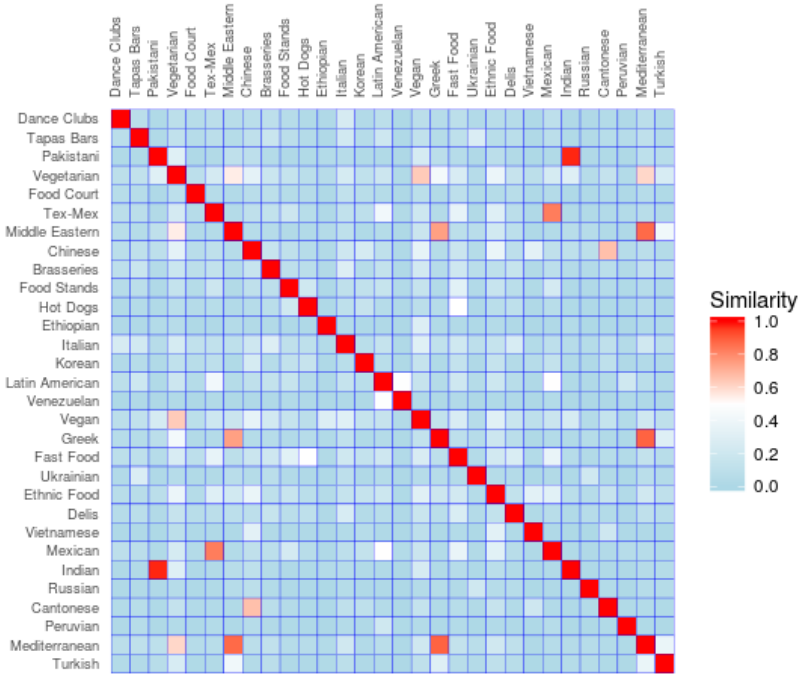


Figure 2. Similarity Matrix from IDF

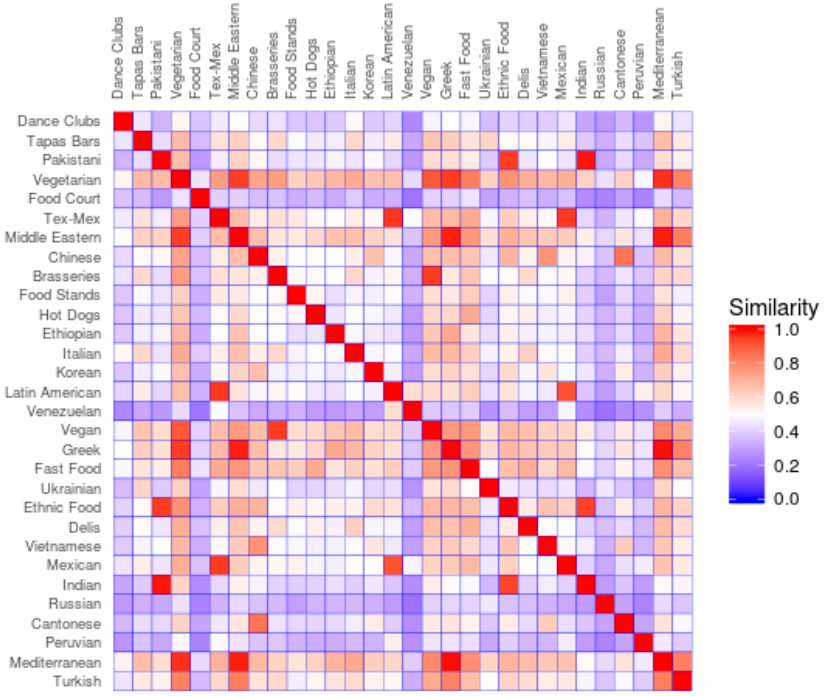


Figure 3. Similarity Matrix from LDA

As you can see, there are a lot of dissimilar cuisines in the noIDF and IDF results because the parameters are not optimal and there is a lot of noise, and it is hard for the program to see the similarities. However, there is quite a number of similar cuisines. Despite the fact that this is a very rough approximation of cuisines, the program still managed to see the similarity between such cuisines as a) Mediterranean, Middle Eastern, Greek, and somewhat Turkish, b) Indian and Pakistani, c) Latin American, Tex-Mex, and Mexican d) Fast Food and Hot Dogs, e) Cantonese and Chinese, as long as some others. The correlation between these cuisines seems to be so strong that you can see it even in the raw text data when no special preprocessing and no IDF is used.

The LDA model managed to smooth many differences out, and a lot of cuisines now have a similarity of around 50% plus there are more similar cuisines, but because of the noise, you can still find it strange that some intuitively different cuisines are shown as having a certain degree of similarity. I made the color of dissimilarity darker (blue instead of light blue), but you can see that the LDA chart still has a lot more red color shades in it than the previous two ones. The similarities noted in the previous paragraph in items a), b), c), d), and e) are also evident in the LDA model results, but these similarities became much more evident because there is more red color in the corresponding squares. Overall, LDA found more similarities between the cuisines. I did not use the black and white palette here (as in the examples from the Task 2 Instructions) because the LDA figure would look too dark, and the first two figures would not have enough contrast to see the similarities found. So I made a decision to use the heat map colors.

Task 2.2

This is where I tried to optimize parameters used in the noIDF, IDF, and LDA models. These same three models are used for comparison with Task 2.1. First of all, in the TfidfVectorizer I changed the maximum document frequency max_df from 0.9 to 0.5 in order to eliminate as many as possible words that are too frequent. I considered a word that occurred in more than 50% of all documents frequent and not interesting. I also changed the minimum document frequency min_df from 1 to 2 because if a

word occurs only in one document, it may be too rare/specific and will not help us with establishing any correlation between the cuisines.

I also used a more complete list of stop words by using stop words from the stop-words and nltk modules as a separate list in addition to the stop word option in the TfidfVectorizer. Also, I included punctuation in the stop word list using the string module. On top of this, I used tokenization from nltk and, naturally, making every word lower case. Also, all non-words were removed using regular expressions. I considered that stemming, lemmatization, or POS tagging may complicate things too much in this generic case, so I didn't use them to save time because the vocabulary is quite big.

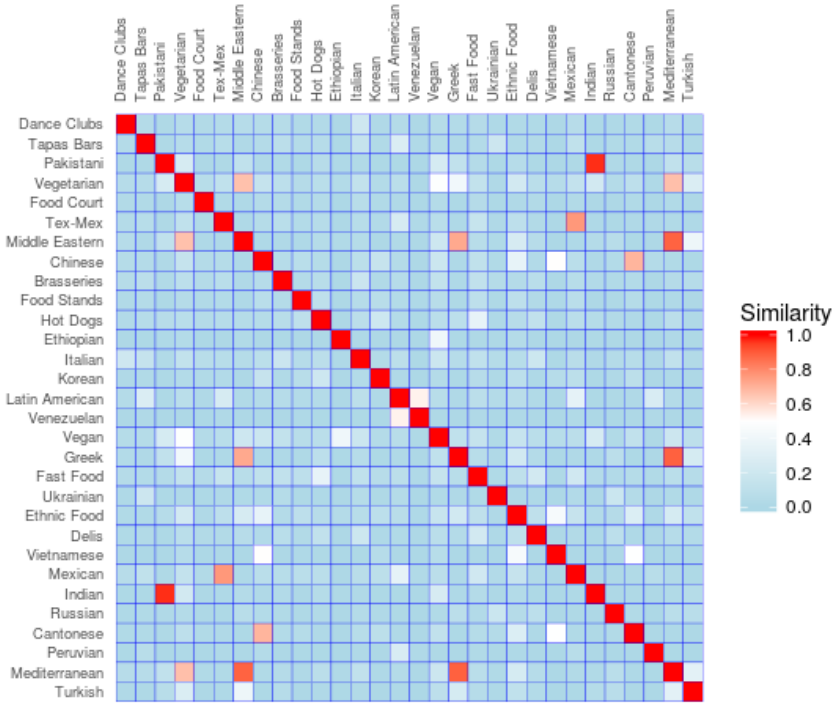


Figure 4. Improved Similarity Matrix from noIDF

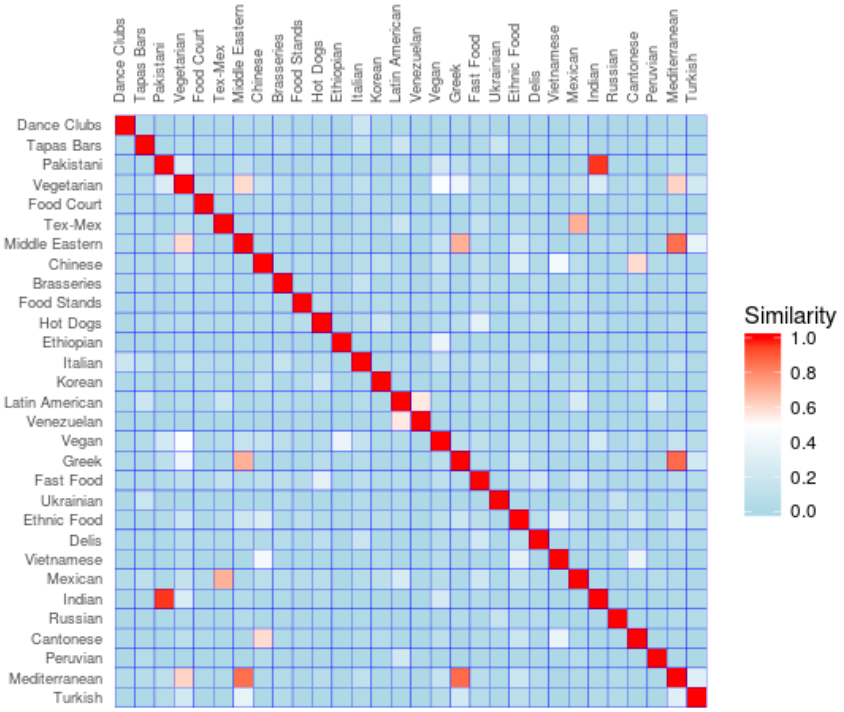


Figure 5. Improved Similarity Matrix from IDF

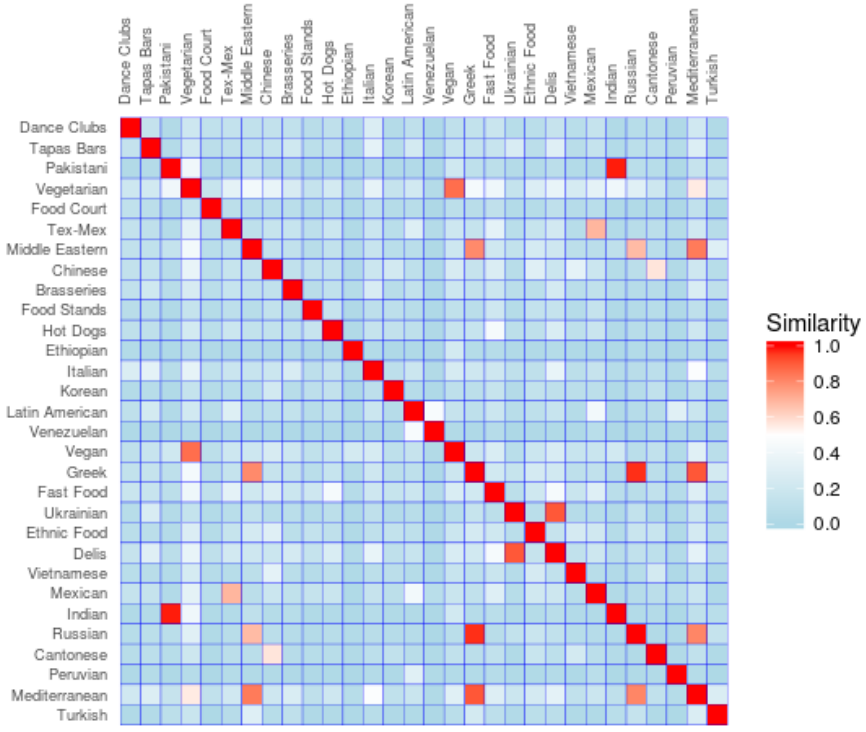


Figure 6. Improved Similarity Matrix from LDA

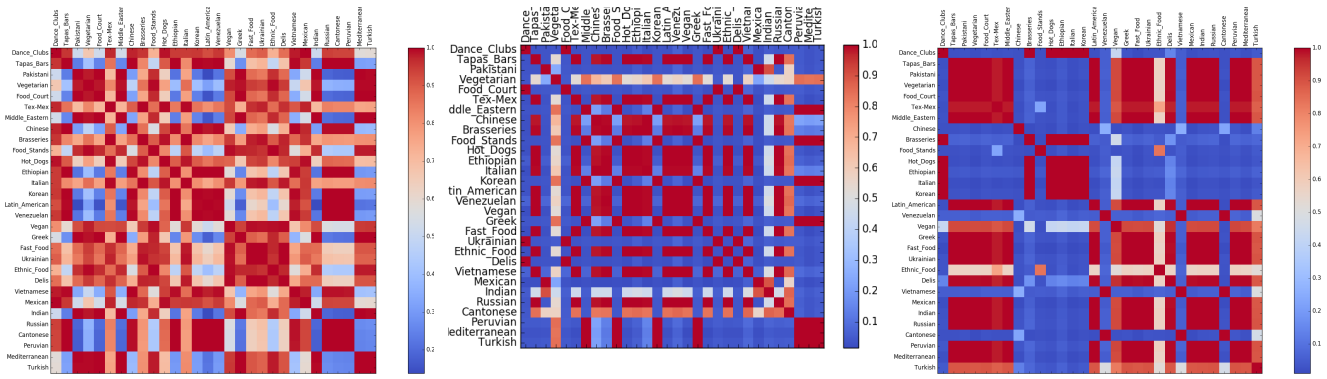
As you can probably see, all the changes that I made did not have a very significant effect on the noIDF and IDF models. I believe this is due to the fact that I initially chose cuisines that are strongly correlated, and even a weak model can see this correlation. As for the LDA plot, it is quite different than in Task 2.1, and now we don't see too many similarities between unrelated cuisines. On the other hand, it became in a way similar to the IDF model, which means that both IDF and LDA models have provided somewhat similar results for Task 2.2.

As usually, you can see varying degrees of correlation between the following cuisines: a) Mediterranean, Middle Eastern, Greek, and somewhat Turkish, b) Indian and Pakistani, c) Latin American, Tex-Mex, Mexican, Peruvian, and Venezuelan d) Fast Food and Hot Dogs, e) Cantonese and Chinese, f) Korean and Chinese, g) Italian, Delis, and Mediterranean, h) Russian, Greek, and Mediterranean (having been to Russia, I can confirm that this is partially so, at least among the dishes that American Russians prefer) , i) Vegetarian and Vegan. It appears that all of these similarities make a lot of sense. Overall, I would rate the results as successful in identifying closely related cuisines.

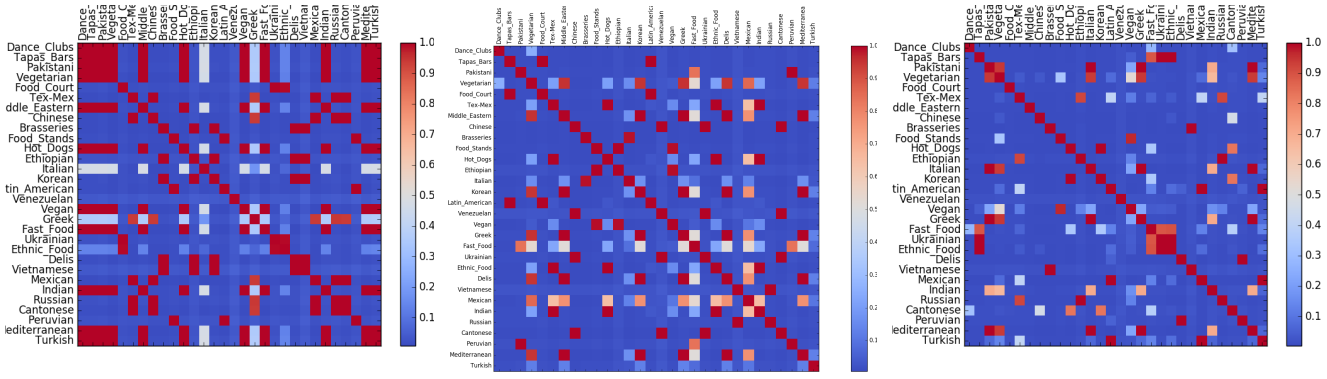
As for the preferred data mining method, I believe that the LDA model provides more flexible and evident results depending on how you tune the parameters.

Task 2.3

My first attempt to cluster was with topic clustering in LDA. LDA calculates the vector representation of each category using the probability that every category belongs to each topic. Then, the cosine distance is used to measure the differences between the cuisines. I tried these numbers of topics: 2, 4, 5, 7, 10, 15, 30. The results are shown in Figure 7 below.



2 topics 4 topics 5 topics



7 topics 10 topics 15 topics

Figure 7. LDA Topic Clustering

It appears that 2 topics is too few because many cuisines are shown as similar. You can see clustering patterns when the number of topics is around 5, including 4 and 7. It is possible to analyze the clusters and draw conclusions. If the number of topics is 10 or 15, it is possible to make judgments about

individual cuisines, but not their clusters. If you examine the generated clusters closer, it looks like clustering with 5 topics makes most sense out of all of these six examples.

The second clustering algorithm that I used was the Ward hierarchical clustering algorithm [3]. Ward clustering is an agglomerative clustering algorithm which means that a pair of adjacent clusters with the minimum distance between them is merged at every stage of the algorithm. I used the scipy.cluster.hierarchy module. The distance matrix was obtained by deducting the similarity matrix from 1. The resulting dendogram is presented in Figure 8.

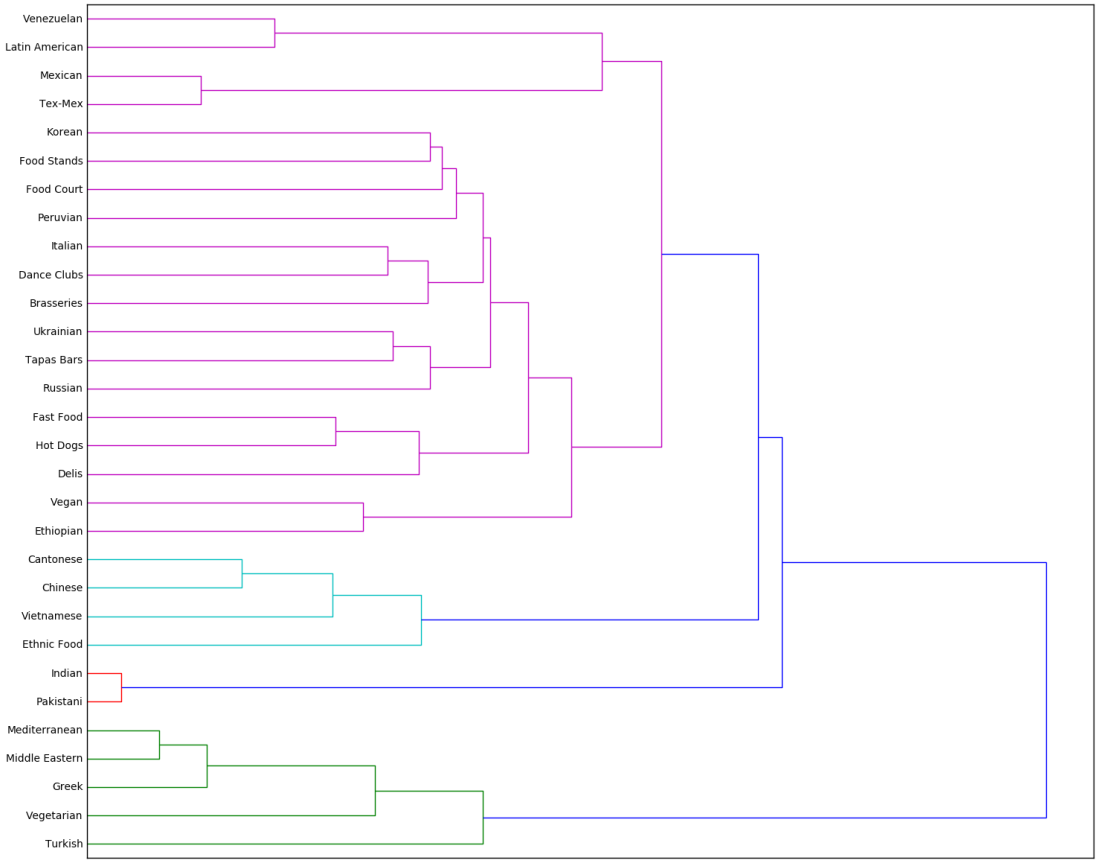


Figure 8. Ward Hierarchical Clustering Dendogram

As you can see, there are some interesting results obtained. Although the overall number of clusters is 2, with the first distinct cluster incorporating the Mediterranean, Middle Eastern, Greek, Turkish, and Vegetarian cuisines which makes perfect sense (the Vegetarian cuisine also had a certain degree of similarity with these cuisines in Task 2.2), the second cluster is further subdivided into a cluster containing the Indian and Pakistani cuisines (very similar according to previously received results) and another cluster which, in its turn, is further subdivided into a hierarchy of related clusters with such similar cuisines as: a) Cantonese, Chinese, Vietnamese, Ethnic Food, b) Fast Food, Hot Dogs, Delis, c) Venezuelan, Latin American, Mexican, and Tex-Mex. Therefore, it is evident that based on the internal structure of the clusters, this method managed to identify the similarity of the main cuisines. The rest of the results are not very precise.

The third clustering algorithm that I tried was K-means from sklearn.cluster. I used the distance matrix for clustering and matplotlib for visualization. In order to show the results in a 2D plot, I used MDS from sklearn.manifold. Pandas Dataframe was used to assist in this process. The results are shown in Figure 9 below.

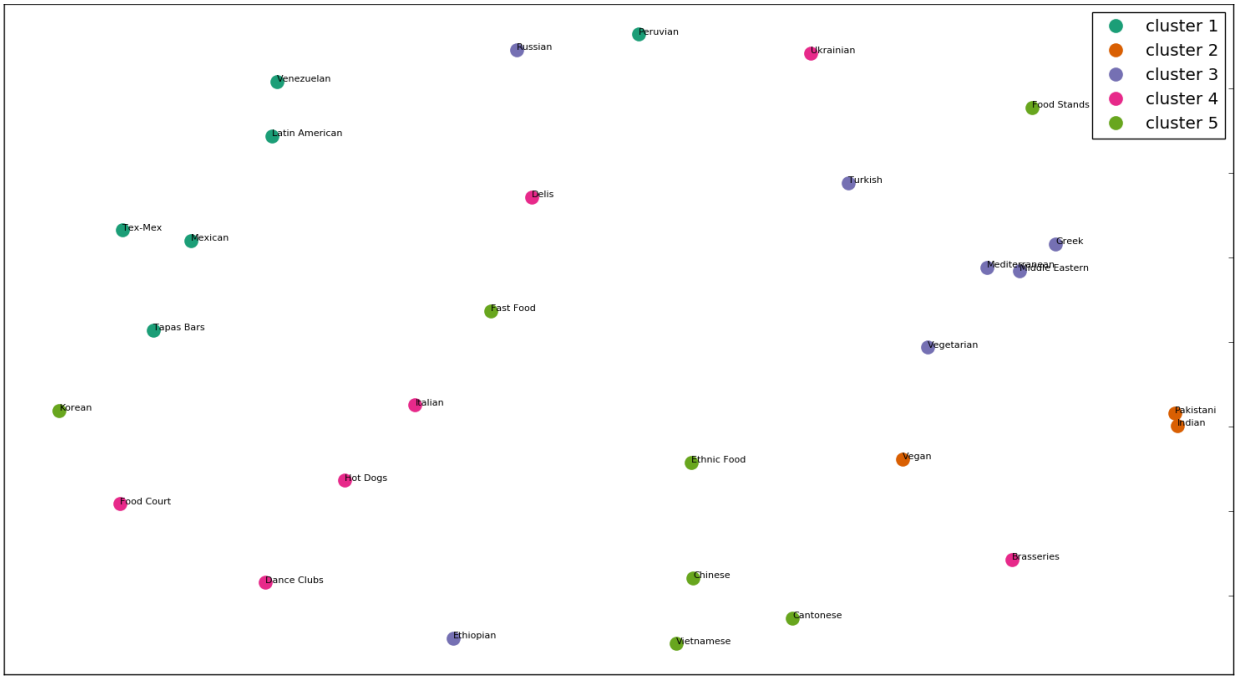


Figure 9. K-Means Clustering

As you can clearly see, the results make sense, and we have approximately the same groups of cuisines: a) Venezuelan, Latin American, Tex-Mex, Mexican, 2) Mediterranean, Middle Eastern, Turkish, Greek, Russian, Vegetarian (as I already mentioned, the American Russian cuisine uses quite a number of dishes that are close to the Mediterranean/Turkish cuisine), 3) Food Courts, Hot Dogs, Delis, Dance Clubs, Brasseries, 4) Chinese, Cantonese, Vietnamese, Ethnic Food (although this category also includes Fast Food and Food Stands for some reason), 5) Vegan, Indian, Pakistani.

In general, one can say that the clustering results received during the implementation of Task 2.3 were successful, and the major groups of cuisines were discovered. Of special use are the Ward's method and K-means clustering as they provide a good visualization of the similarities found. However, K-means turned out to be more efficient in providing quality clusters and made less mistakes compared to the Word's method.

References

1. Document clustering with Python: http://brandonrose.org/clustering
2. Calling R from Python: https://sites.google.com/site/aslugsguidetopython/data-analysis/pandas/calling-r-from-python AND https://www.r-bloggers.com/ggplot2-in-python-a-major-barrier-broken/
3. Ward's method: https://en.wikipedia.org/wiki/Ward%27s_method