Clustering cities by venues categories

By Marzio Lanzoni, 13/01/2021

1 Introduction

1.1 Background

Big cities evolve very quickly, following economic perspectives, lifestyle trends and political scenarios.

Common thinking tends to assume similarities among cities based on geographical proximity or historical connections, but nowadays many factors affects cities, and may turn common thinking and traditional assumptions outdate and very misleading.

It would be interesting to build a machine learning approach to constantly update the similarity among big cities all around the world based on actual and update characteristics.

1.2 Problem

Leveraging near real-time data like the ones provided by Foursquare APIs allows a continuous update of how a city is changing, and this is something that conventional approaches like surveys and inventories cannot accomplish: surveys need to be planned, executed and published, and that is why they risk being outdate before becoming effective.

Clustering big cities through a machine learning approach makes possible to sense changes happening in the city even (I would say primarily) when these are difficult to foresee and driven by patterns complex to understand.

1.3 Target

Clustering big cities based on the kind of entertainment services they offer, or on the ethnic cuisines mostly covered by their restaurants, or other categories of Foursquare's venues which are supposed to reflect the mood of the city and the overall spirit you can find in it: this could be a valuable tool for people aiming to travel and who are looking for experiences and feelings they previously had visiting other cities.

2 Data acquisition and cleaning

2.1 Data sources

2.1.1 Simplemaps (locations)

As a reference for cities locations, we will use "Simplemaps" database[1]. It is a public database of the world's cities and towns, built it using authoritative sources such as the NGIA, US Geological Survey, US Census Bureau, and NASA. The database is constantly up to date (last refresh November 2020). The basic (free) version of the database contains data of about 26 thousand prominent cities (large, capitals etc.) and will be used to extract the location of some cities (see 0 for data selection criteria) to feed Foursquare's APIs.

2.1.2 Foursquare (venues)

Version 2 (namely 20180605) of the Foursquare's APIs will be used.
Venues Category method will be used to get a full list of possible venues categories available.
Venues Search method will be used to get venues located in the downtown area of the selected cities.

2.2 Data cleaning

2.2.1 Simplemaps data

From Simplemaps database we will extract a subset of cities with medium-high population (2.5M+ habitants), which are supposedly more tagged on Foursquare and have a considerable number of venues.

We will exclude Chinese cities, because they are much more populated than other cities in the world: keeping them would increase too much the dataset or would exclude cities from almost all other countries. Of each city retained in the set I will keep only the name and the location (latitude and longitude).

2.2.2 Foursquare data

Foursquare's APIs allow to get venues of countless cities all around the world.

Each venue returned by the APIs is categorized in a very detailed manner, since more than a thousand categories can be used. Hopefully, each venue's category belongs to a hierarchical structure made by several levels, the upmost of which (level 0) is made by the following items:

- Arts & Entertainment
- College & University
- Event
- Food
- Nightlife Spot

[1] https://simplemaps.com/data/world-cities

- Outdoors & Recreation
- Professional & Other Places
- Residence
- Shop & Service
- Travel & Transport

For example, there are 265 possible categories related to the topic 'Food', and they are nested up to 5 levels. Of course, such a categorization would reflect in having venues fragmented in too many categories with very low frequency each, while it seems obvious that it would be much more meaningful to aggregate them in order to have more occurrences in each family and a lower segmentation.

To this end, the category by which each venue will be tagged will be transcoded by its 2nd level parent in the hierarchical structure said above. By that, the 265 possible categories of the topic Food will be transcoded in the following, much more meaningful and representative, 21 items:

- African Restaurant
- American Restaurant
- Asian Restaurant
- Caribbean Restaurant
- Dessert Shop
- Eastern European Restaurant
- French Restaurant
- German Restaurant
- Greek Restaurant
- Hawaiian Restaurant
- Indian Restaurant
- Italian Restaurant
- Jewish Restaurant
- Latin American Restaurant
- Mediterranean Restaurant
- Mexican Restaurant
- Middle Eastern Restaurant
- Russian Restaurant
- Spanish Restaurant
- Turkish Restaurant
- Ukrainian Restaurant

In addition to this, other cleaning strategies will be applied to increase the overall meaningfulness of the features:

- Only venues found in a radius of about 500[2] meters around the centre of each city will be kept, based on the assumption that venues close to the centre are most significantly representing the city, while all the cities get similar the more you step away from the centre.

[2] See the notebook for the actual implementation.

- The number of venues will be restricted to 100 venues per city, because this is the limitation of the free of charge Foursquare license.
- Cities for which the returned number of venues is below a given threshold (about 40^2 venues per city) will be dropped from the dataset, to ensure that the venues number is large enough to meaningfully represent the city.

2.3 Feature selection

Foursquare's APIs will return different many venues for each city, and each venue will belong to one category. So, the dataset will be made by many rows (venues) for each city, with only one category per row.

First, the categories will be transcoded according to the strategy described in 2.2.2, to avoid too much segmentation.

Then, we will apply a **one-hot** encoding to the dataset, in order to get a new dataset where each city is a row, the columns are the categories got from the union of the categories returned for all venues of all cities, and each cell contain the number of venues associated to one category (column) for one city (row).

Since the number of venues per city is different, we need to normalize the number of venues per category per city by the overall number of venues for that city.

This will return the frequency by which each category occurs for each city. Such a dataset is finally consistent, and the categories frequency will be used as features to feed a clustering algorithm.

By purpose, we will exclude from the features every information related to geographical connections (for example coordinates), because one of the main goals of the project is to refute the assumption that proximity implies similarity and let the actual venues categories shape the true similarity among cities.

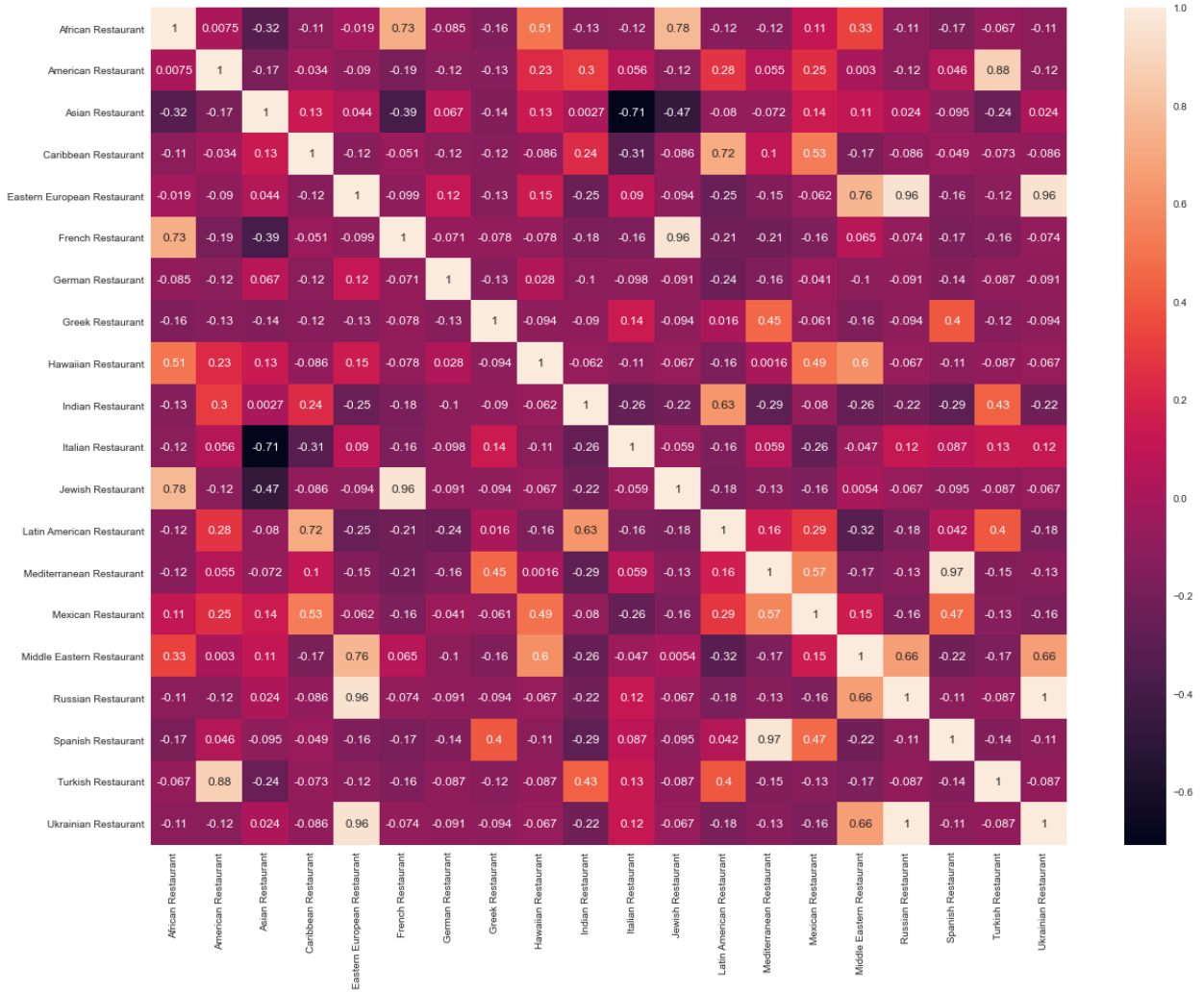
2.4 Exploratory Data Analysis

Before feeding with features any clustering algorithm, it is best practice to explore the dataset, in order to understand if there is some evident issue about it, or if there are correlations suggesting a further cleaning of the dataset.

2.4.1 Correlations

Given the very nature of the features, we do not expect our features to show strong correlations. In fact, they are directly related to the category to which each venue belongs, and we do not see any reason for strong correlations among venues[3].

Here is a heatmap of the correlation matrix among the categories used as features.



As expected, correlations are usually small. The few cases where correlation is larger than 0.75 can be always explained because of the intrinsically similar type of cuisine, which can drive to ambiguous

[3] The only correlation mechanism that could possibly occur is that, based on the success of a specific cuisine, similar cuisines could arise in the neighbour or, at least, in the same town.

categorization. For example, Eastern European Restaurant, Middle Eastern Restaurant and Russian Restaurant are intrinsically similar, which may cause a restaurant to be tagged one way or another. The same apply to Caribbean Restaurant and Latin American Restaurant.

2.4.2 Normalization

Despite data must be usually normalized by means of specific scaling techniques, the way the dataset is built makes it useless in the case under study. In fact, once performed the one-hot encoding and the divided the resulting values by the number of venues per city, the resulting values are already normalized and made insensible to the different number of venues per city[4]: by definition, each feature in the final dataset belong to the range [0,1].

2.4.3 Variance

By means of a boxplot of the features (Figure 1), it can be seen how some features have mean value lower by orders of magnitudes with respect to others. The very low occurrence features shall be dropped from the dataset because they do not carry any information and only make the model less steady.

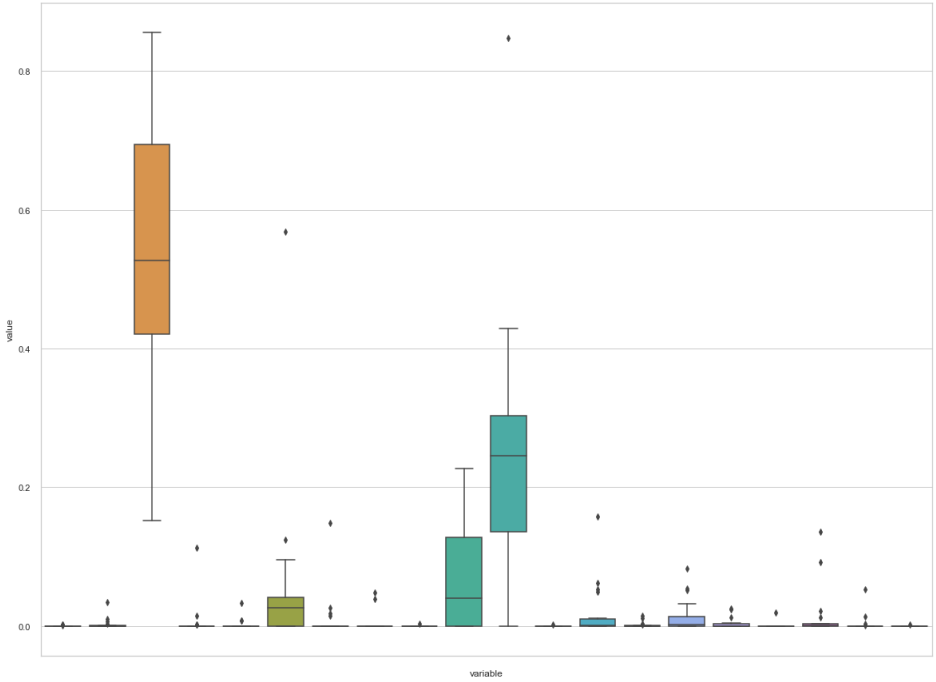


Figure 1: boxplot of the complete set of features.

[4] In 2.7 we will briefly discuss about a possible dependency on the number of venues, but this is eventually related to the meaningfulness of the venues themselves, not to a missing normalization procedure.

In 2.5 we will describe differences encountered when carrying on the analysis using the entire set of features and when just using a subset of features with higher occurrence (namely those with mean > 0.01).

In the following, we will refer the two dataset respectively as “complete” and “restricted”.

Here is the boxplot of the only higher occurrence features.

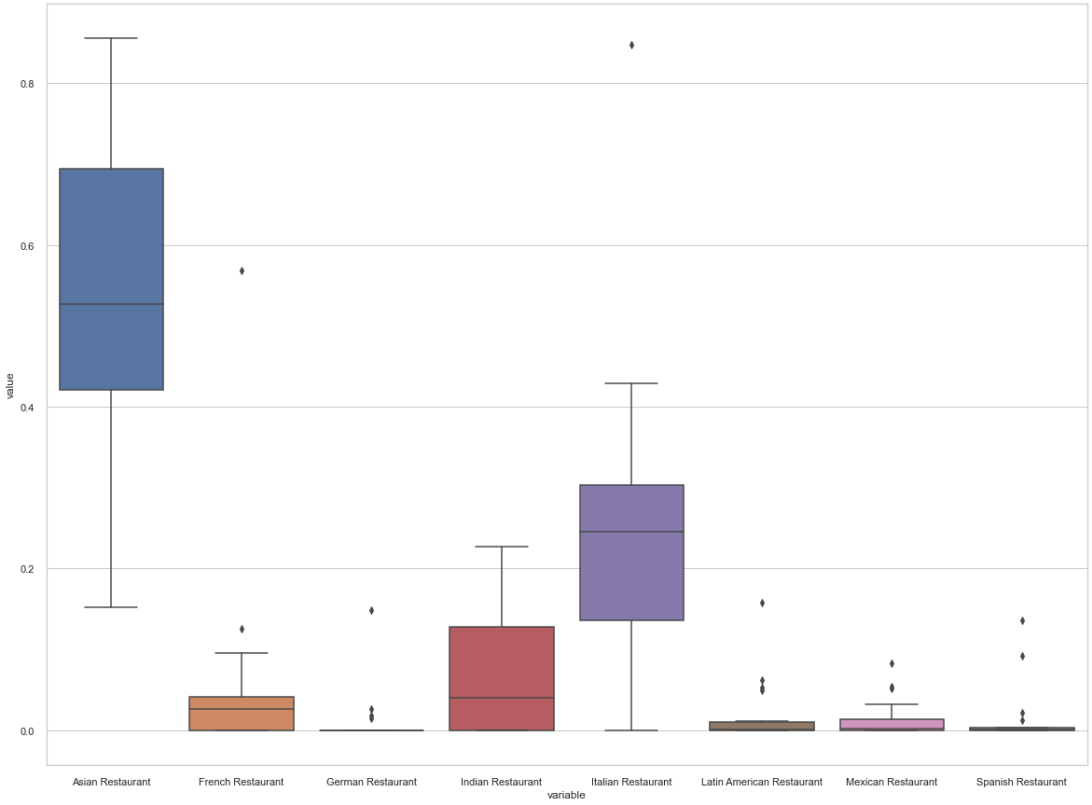


Figure 2: boxplot of the subset of features with Mean > 0.01.

Here is a tabular description of the features with Mean > 0.01 statistics:

	Asian	French	German	Indian	Italian	Latin American	Mexican	Spanish
count	18	18	18	18	18	18	18	18
mean	0.5399	0.0599	0.0115	0.0642	0.2463	0.0191	0.0147	0.0153
std	0.2022	0.1314	0.0351	0.0727	0.1902	0.0403	0.0242	0.0371
min	0.1524	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25%	0.4207	0.0000	0.0000	0.0000	0.1362	0.0000	0.0000	0.0000
50%	0.5269	0.0260	0.0000	0.0403	0.2453	0.0008	0.0023	0.0012
75%	0.6946	0.0417	0.0000	0.1275	0.3034	0.0096	0.0135	0.0033
max	0.8553	0.5680	0.1486	0.2266	0.8476	0.1583	0.0827	0.1362

2.5 Predictive Modeling

2.5.1 Strategy by steps

The problem is a classical unsupervised clustering problem, and we will apply both K-means and Agglomerative algorithms to find if significant differences arise by using these two methods.

For both algorithms, the only parameter to be tuned is the number of clusters the algorithm shall use. To find the best value for such a parameter, we will leverage the KElbowVisualizer library, which implements the "elbow" method to select the optimal number of clusters by fitting the model with a range of values for K. If the line chart resembles an arm, then the "elbow" (the point of inflection on the curve) is a good indication that the underlying model fits best at that point. In the visualizer, the "elbow" will be annotated with a dashed line.

By default, the KElbowVisualizer scoring parameter metric is set to Distortion, which computes the sum of squared distances from each point to its assigned centre. However, two other metrics can also be used with the KElbowVisualizer: Silhouette and Calinski Harabasz[5].

Once discovered the best value for the number of clusters, both K-means and Agglomerative algorithms will be executed, to check if they return similar results.

The Silhouette, Calinski Harabasz and Davies Bouldin scores of the final clustering will be computed as an overall measure of the clustering performance.

2.5.2 Complete vs Restricted set of features

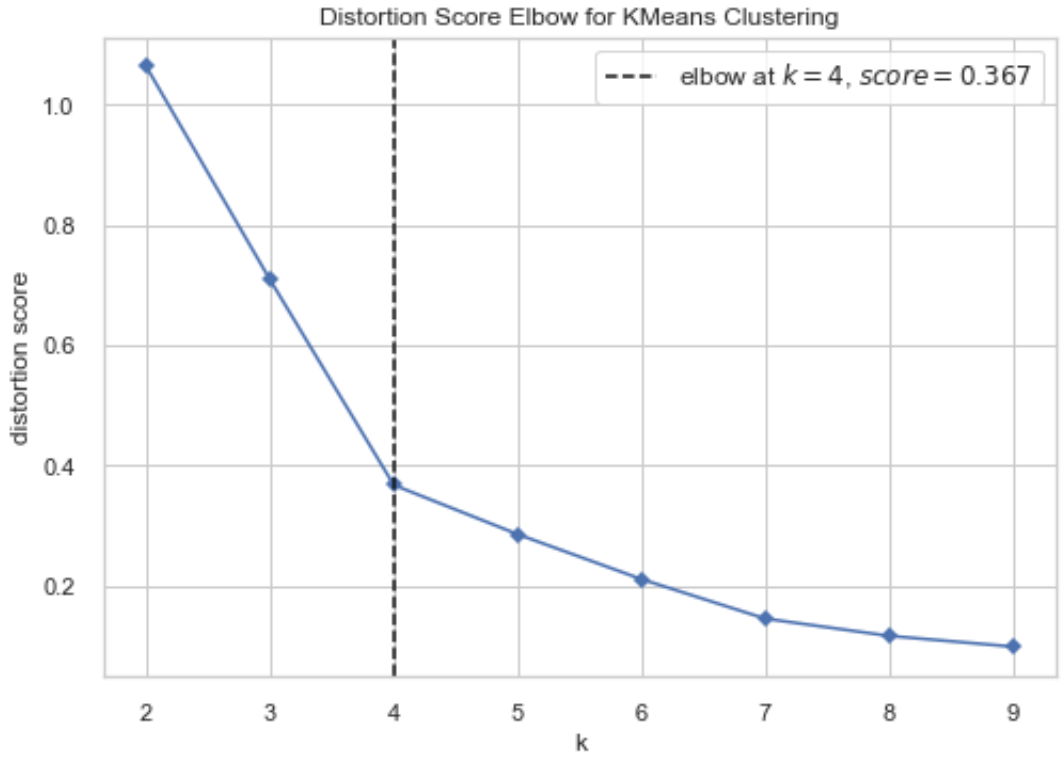
When running on the complete set of features to find the optimum K value, the KElbowVisualizer function fails to converge when using the Silhouette metric, while using Distortion and Calinski Harabasz the KElbowVisualizer function behaves the same way.

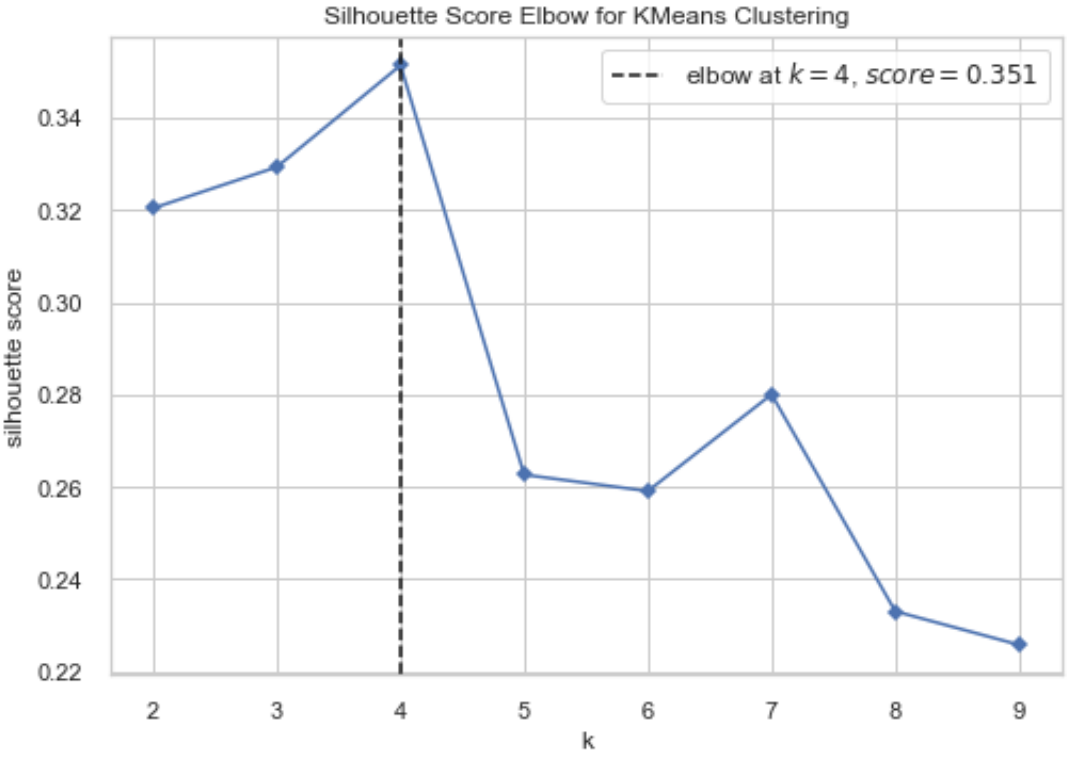
For this reason, here after we will be carrying on the analysis on the restricted dataset, meaning the one made only by features with Mean greater than 0.01.

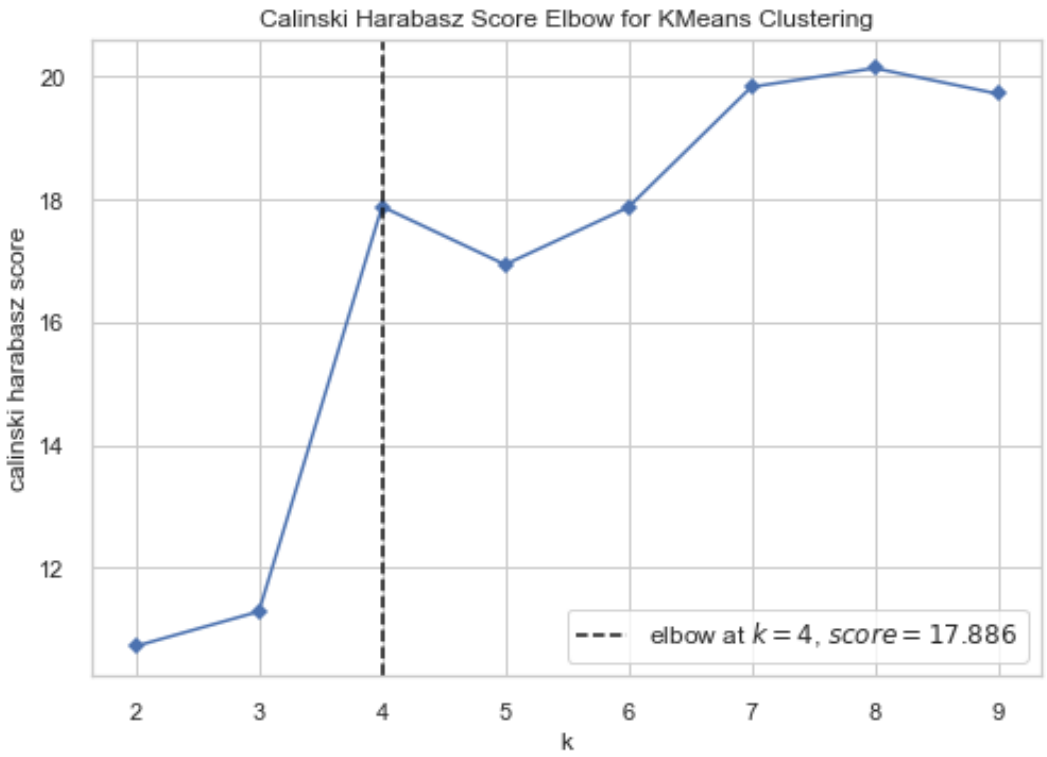
2.5.3 Tuning the algorithm parameters: number of clusters

Since both K-means an Agglomerative algorithm only require tuning one parameter, namely the number of clusters to be used, we will decide this value plotting Distortion, Silhouette and Calinski Harabasz score obtained from KElbowVisualizer library on the dataset described as "restricted" at 2.4.3.

[5] https://medium.com/@haataa/how-to-measure-clustering-performances-when-there-are-no-ground-truth-db027e9a871c for a comprehensive explanation of these metrics definition.


Distortion Score Elbow for KMeans Clustering
elbow at $k = 4$, $score = 0.367$
distortion score
1.0
0.8
0.6
0.4
0.2
2
3
4
5
6
7
8
9
k


Silhouette Score Elbow for KMeans Clustering
elbow at $k = 4$, $score = 0.351$
silhouette score
0.34
0.32
0.30
0.28
0.26
0.24
0.22
2
3
4
5
6
7
8
9
k



Each metric suggests the same value for the best K, K = 4.

We will not plot the same curves for the Agglomerative algorithm since they look the same as the curves obtained running K-means.

2.5.4 Final algorithm selection and scoring

Running both K-means and Agglomerative algorithms with K = 4 on the dataset described as “restricted” at 2.4.3., we get the same Silhouette Score = 0.3514.

Such following predictive scores:

- Silhouette Score: 0.3514
- Calinski Harabasz Score: 17.8863
- Davies Bouldin Score: 0.5610

2.6 Conclusions

Applying the Agglomerative algorithm to the dataset described in we get the clustering here below shown on a map by means of Folium library.

For a better understanding, here is a tabular version of the clustering obtained:

Red Cluster
Barcelona
Berlin
Birmingham
London
Madrid
Manchester
Moscow
Washington

Blue Cluster
Hong Kong
Montréal
New York
Seattle
Tokyo
Ōsaka

Cyan Cluster
Paris

Yellow Cluster
Rome

Table 1: final clustering (Agglomerative algorithm with K = 4).

Here are some intuitions for the clustering obtained:

- All the European cities have been clustered altogether, exception made for Paris and Rome.
- Paris and Rome, despite European cities, have been clustered separately and they constitute a cluster each, probably because French and Italian cuisines are so renowned and valuable at commercial level, so attractive as brands, that the capitals of the respective countries likely have a huge number of national cuisine restaurant, much higher than other capitals with respect to their national cuisine.

- USA and Japan cities have been clustered together maybe because they tend to have more a cosmopolitan characterization than a national or local one.

Despite we can find a reasonable interpretation of the clustering results, still the scores shown in 2.5.4 suggest that clusters are quite overlapping, with samples quite close to the decision boundary of the neighbouring clusters. Strategies suggested in 2.7 could possibly drive to more dense and well separated clusters.

2.7 Future directions

Possible future directions for the presented analysis are mostly related to stepping back to the various assumptions done during the report and studying how different assumptions change the results.

Radius

We set a 700 metres radius within which to search for venues. The general idea driving this choice is that representative venues are closer to the downtown. Even confirm this assumption, which seems solid, the definition itself of "close to the downtown" likely depends on the size and topology of the city, which suggest varying the radius based on the city physical properties.

Venus categories aggregation

At a first glance, the way Foursquare's categories are hierarchically organized is questionable, meaning that categories located at the same level of the hierarchy have different meaningfulness. Trying a custom aggregation could result in a better representation of the culinary profile of the cities.

This approach is also suggested by the data exploration performed at 2.4, by both the correlation and the variance analysis.

- The correlation analysis shows that there are categories highly correlated.
- The variance analysis suggested to drop some features because they have very low occurrence.

Both issues will be mitigated by trying a different aggregation, where categories highly correlated shall be merged, and categories with very low occurrence shall be merged to very similar categories in order to get a significant occurrence.

Expanding Categories

For sake of simplicity, the presented analysis is based only on "Food" main category, while other main categories (like Arts & Entertainment, Nightlife Spot, Outdoors & Recreation) could help profiling the cities.

Number of venues

The dataset used is made by cities with very different number of venues, likely due to different population and different touristic appeal (it would be interesting to check it out). Also, for some city the number of venues reached the 100-limit imposed by the free Foursquare license, which may result in a bias because the overall number of venues in the downtown radius could be much larger and the returned venues are not necessarily the most representative. Unlocking Foursquare license and trying to use cities with more similar number of venues is a possible direction of investigation.