

CITY CLUSTERS

NON-SPATIAL PROXIMITY

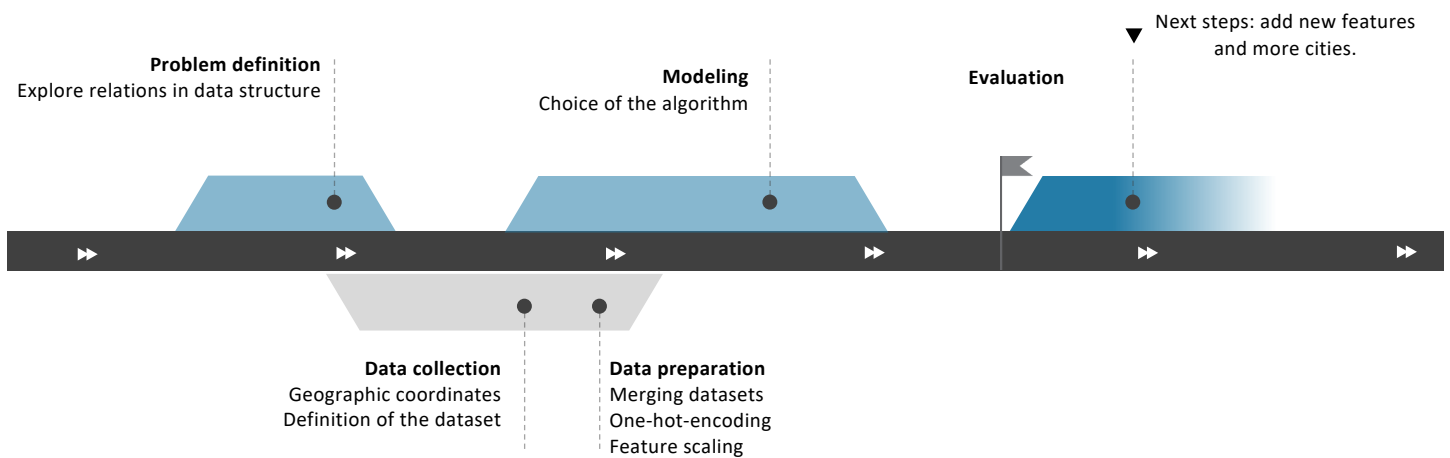
Background: Classification and segmentation of complex systems into simpler categories are ubiquitous to technological advance and a straightforward way to make better customer services, recommend more accurate products and narrow down

options for our limited brain capacity for decision making, improving satisfaction with the choice by making the process easier. To illustrate one such application of a classification method this project seeks patterns that can be used to classify entire cities based on its venues.



Klebert Toscano de S. Cintra
<https://github.com/KTSC>

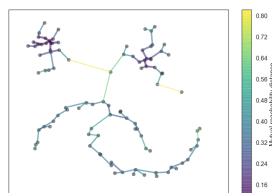
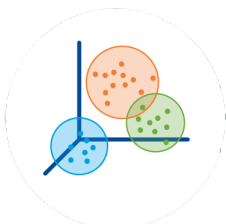
METHOD:



TOOLS



...and the HDBSCAN package



LINEAR
ALGEBRA



DATA WRANGLING



API Integration

WHAT CITIES ABROAD HAVE THE COMBINATION OF FEATURES TO MAKE YOU FEEL LIKE HOME?



TARGET AUDIENCE – USE CASES

TRAVEL AGENCIES

TOURISTS

RECOMMENDATION

STUDENTS

Data Science

SPATIAL ANALYSIS


PROFILING

CUSTOMERS

PRODUCTS

Contents

1. Introduction
 - 1.1 Problem statement
 - 1.2 Interest and Target Audience
2. Data Acquisition and Processing
3. Data Modeling



City Clusters: Non-spatial Proximity

Klebert Toscano de S. Cintra, M.Sc.

1. Introduction

1.1 Problem Statement

Suppose a tourist from Sao Paulo wants to spend some time in a city abroad. He thinks having to process phrases in a foreign language is enough work for his vacationing brain so he wants to find cities that can make him feel as “home” as possible. He is concerned about the diversity and availability of transportation, the services around his residence that he is used to hire, the entertainment he enjoys, etc. The present work proposes to find a systematic way to assess the existence of a Sao Paulo “twin” in the United States, and check if that country is a good choice of destination based on this criterion.

The rationale is translational for other business problems where one has a great number of features of unknown structure or relationship among the variables.

1.2 Interest and Target Audience

This type of analysis is of interest of travelers who seek similar or contrasting destinies when compared to their city of origin. With a few changes It is also applicable for grouping real estate by its multidimensional similarities that are not obvious nor easy to visualize or for recommendation systems in general.

2. Data Acquisition and Processing

For information about local businesses, venues, activities, healthcare, transportation, etc. we will use the Foursquare API location. For comparison we will use the list of cities available on Wikipedia at the URL https://en.wikipedia.org/wiki/List_of_United_States_cities_by_population. Those cities have been listed there for being the largest cities in population count in the US.

The data must be downloaded and combined into one table, to be one-hot-encoded for category of venues and summarized for comparison.

The API calls used for this analysis aimed to explore venues in a radius of 10000 meters of the points of interest, that is the central point of the cities compared to the location of residence of the subject for this simulated business problem. This point for comparison is Avenida Paulista, Sao Paulo, Brazil.

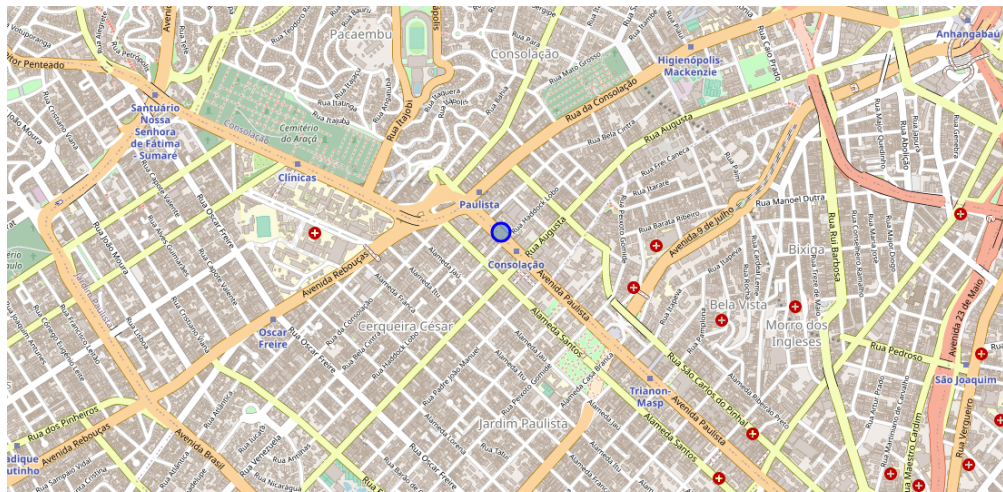


Figure 1: Map of the point of reference for this analysis, with request for all venues in the radius of 10 Km for all points.

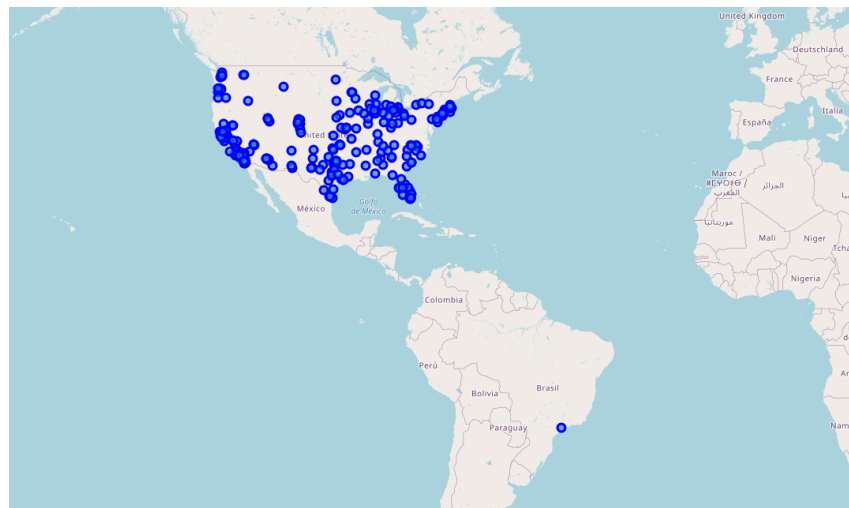


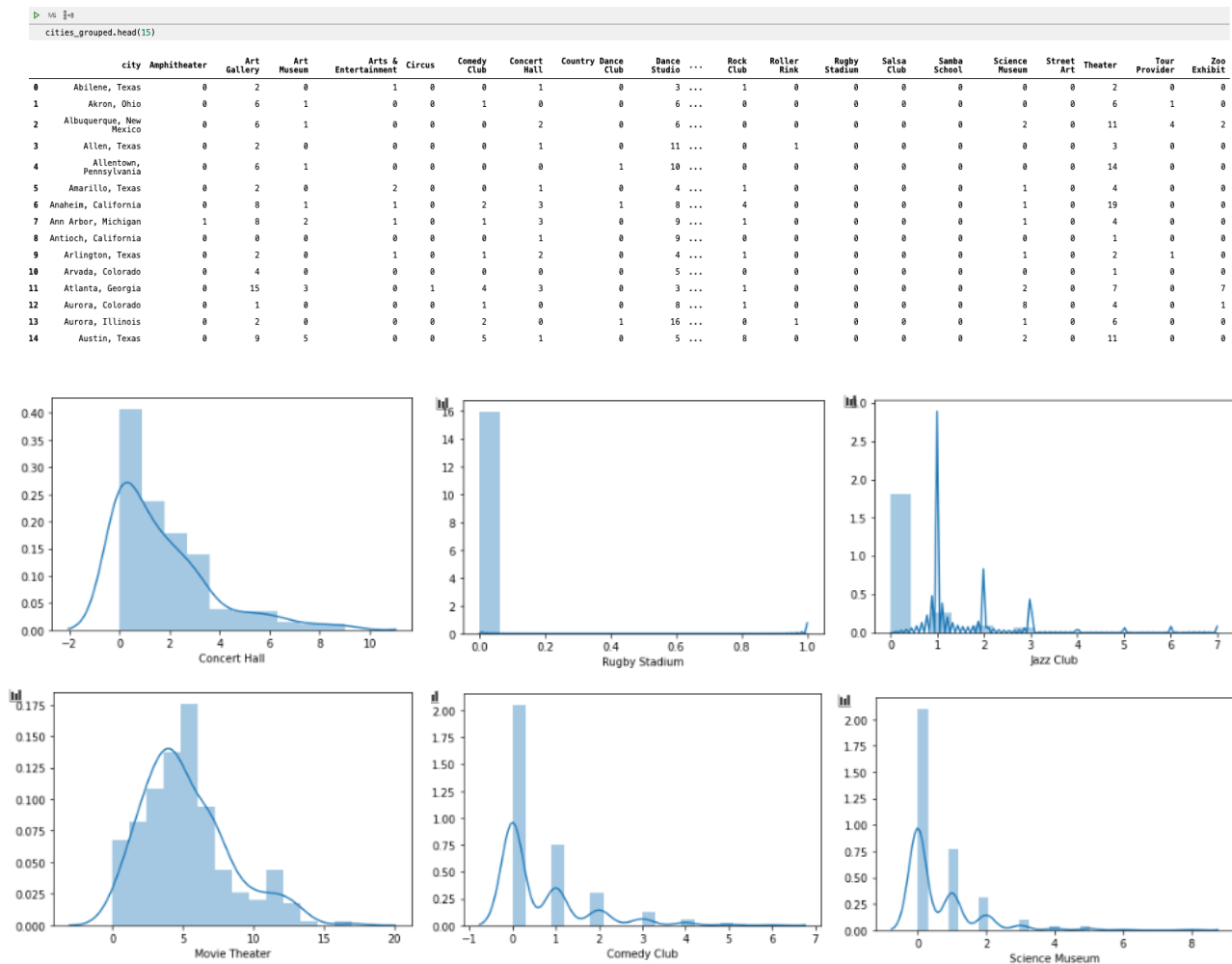
Figure 2: Map of the 280 cities considered for this analysis.

The city names were acquired from the table featured in the aforementioned URL, and latitude and longitude coordinates for each city name were consulted using the geolocator Nominatim.

Those coordinates were used to make API calls to the Places API by Foursquare, requesting all locations, up to the API's limit, of venues of all categories except restaurants. The reason for that is that restaurants' categories are at the same time the most numerous outputs of the API and the less differentiable venues among cities, since all major cities in the world feature international cuisines. Including food related venues in the calls would only compete with the limited 100 responses per location requested.

The outputs were then joined in one table and the categories were one-hot-encoded, including the population for each city, as it is also relevant as a feature that characterizes a city. Because the API's output doesn't provide the same number of venues for all cities, it seemed more reasonable to use the sum of all venues for each category instead of the average of that specific category per city. The resulting one-hot-encoded table and the distribution of some of the features are shown below:

Figure 3: Table with the One-hot-encoded data set and the histogram of 6 different features of the data.



The features histograms are clearly skewed, and don't resemble a normal distribution. Each feature represents a different dimension for the data points, and our goal is to find similarities between these points in

order to associate them in clusters. This is a process of partitioning these data objects into groups according to some similarity or dissimilarity – in other words, a distance criterion. There are 42 dimensions in total, and the more dimensions we add, the more equidistant the points become, making the distance meaningless. In order to reduce the dimensions taken into account we will use a dimensionality reduction technique called Principal Component Analysis. It consists of finding a way to cross the higher dimensional planes in order to represent most of the variation of the data points present in that higher dimension, but using a lower dimensional space. The figure below shows points in a 5-dimensional space being “reduced” to only a plane with 2 dimensions. Each new plane that crosses the multidimensional space must be perpendicular to the previous one, that is called orthogonal plane.

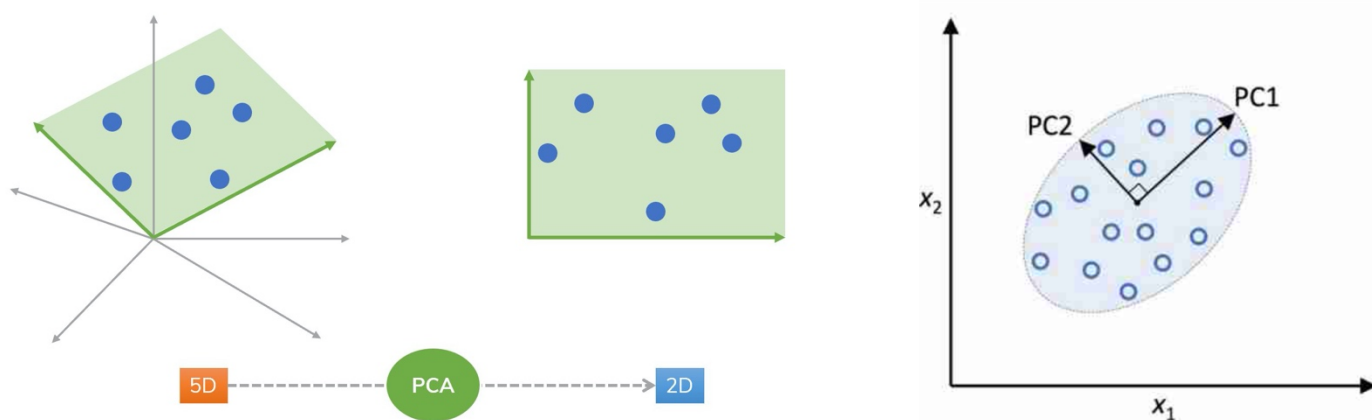


Figure 4: Illustration of the dimensionality reduction performed with the Principal Component Analysis method (left). The plot on the right shows an illustration of the orthogonality between two components of the analysis, named PC1 and PC2.

Because PCA is trying to find components that maximize the variance in the data, it is important that the data has similar variance to start. That is a problem we need to remediate because as can be seen in figure 3 the range of the data and their distribution is not proportional neither normal. This can be solved by the Robust Scaling method that is based in the quartiles of the data, well suited for data that is not normal, and the method is not sensitive to outliers.

After scaling and searching for hyperplanes with PCA we find that 11 components are enough to explain 90% of the variation in the data – which is much better for our analysis than the 42 initial dimensions.

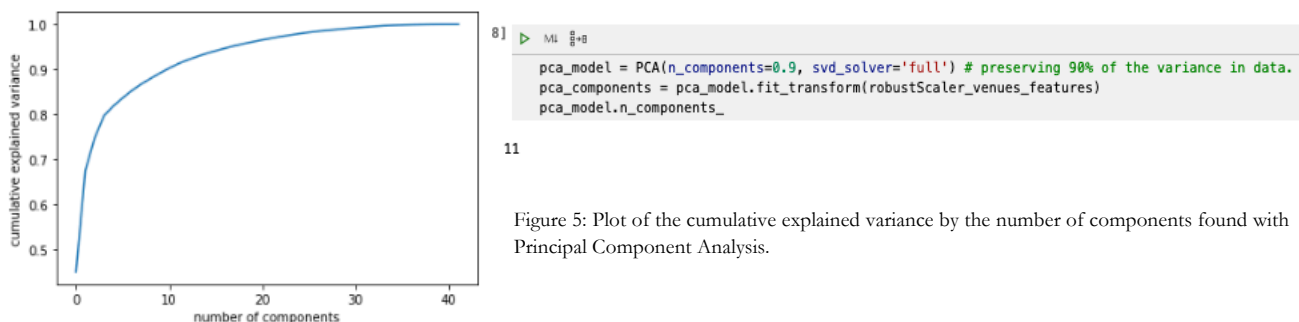


Figure 5: Plot of the cumulative explained variance by the number of components found with Principal Component Analysis.

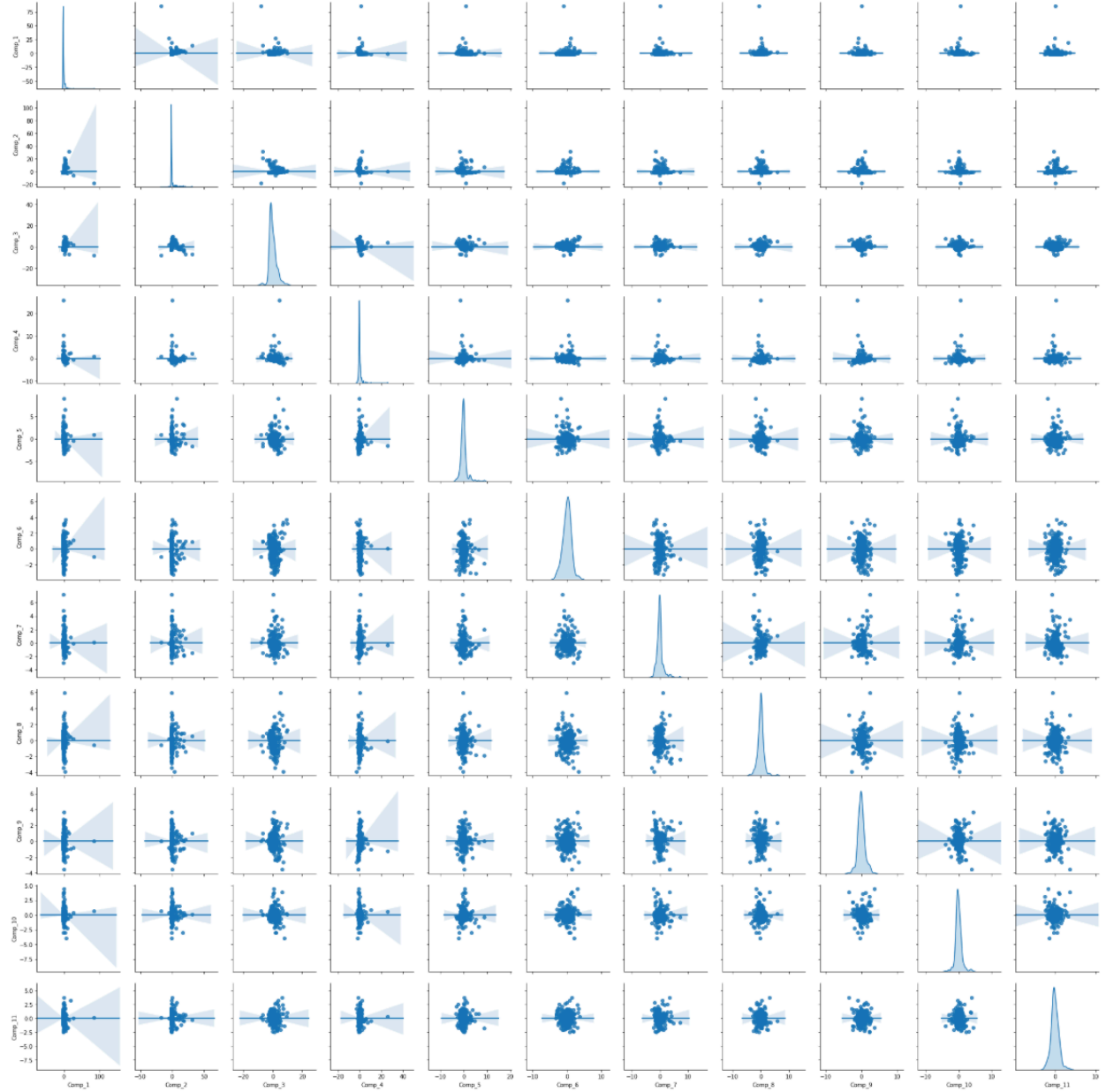
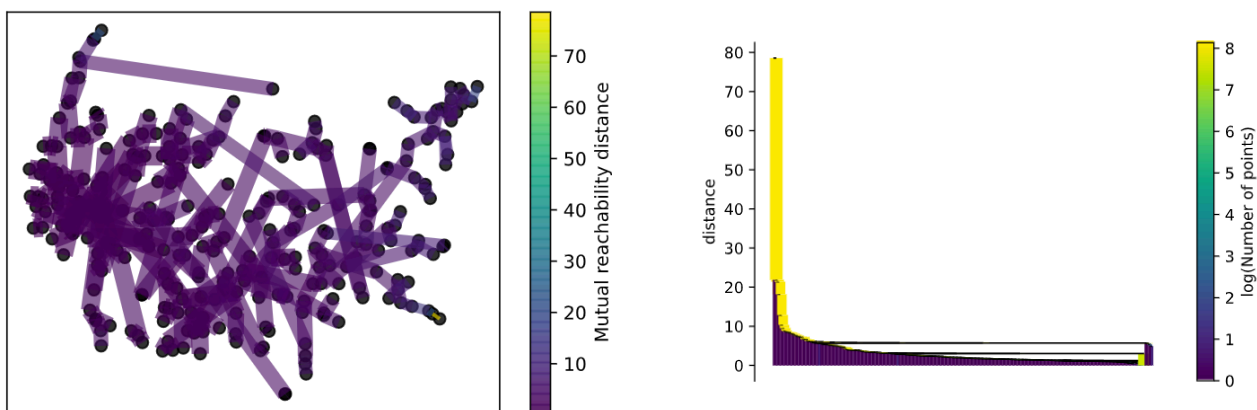


Figure 6: Plot of all 11 components found with PCA. The pair plot can evidence correlations between variables, which can't be found among these 11 components.

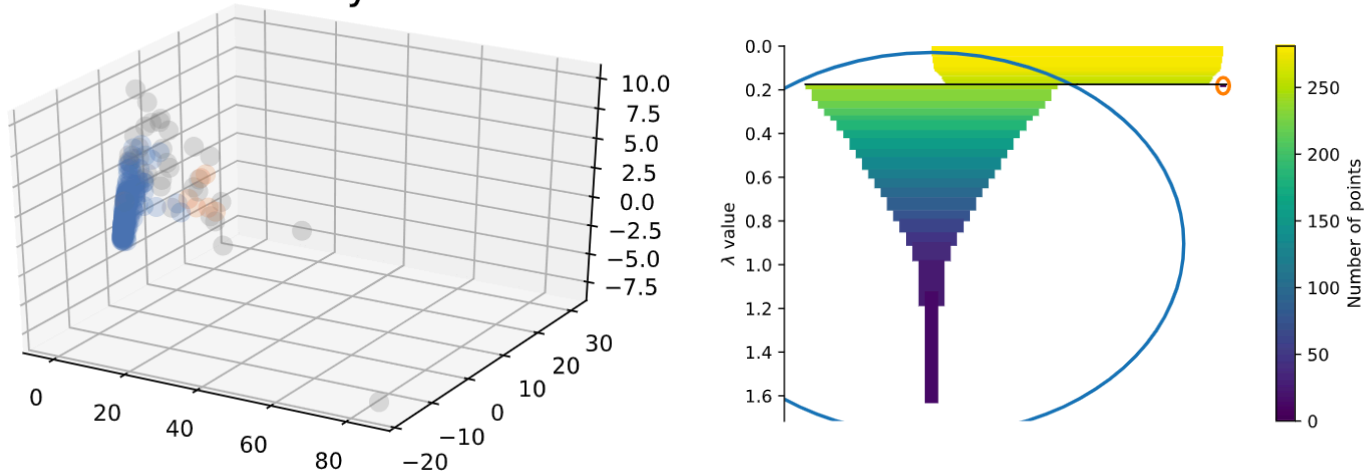
3. Data Modeling

With these components as our new dimensions, the data points can be measured and compared for distances with a clustering algorithm. In order to make no assumptions regarding the shape of the clusters or their densities, the Hierarchical Density Based Spatial Clustering of Applications with Noise (HDBSCAN) seem to be the option of choice. Instead of using it for spatial clustering in the geographical area, it is applied here in

the 11-dimensional space to search for clusters in the distances in the 11 dimensions that represent the 42 original features of the venue's categories of the cities in comparison here.



Clusters found by HDBSCAN



The figures above show that the visualization of the 11-dimensional space in 2D doesn't evidence a clear cluster. Nonetheless, the analysis of the dendrogram shows 2 distinct clusters, rendering the remaining as “noise”, which is assumed by the model. The lower left figure shows the 2 clusters in orange and blue, and the summation of the edges between these data points are stacked in the visualization to the lower right, where the clusters are circled with the respective colors.

The original problem is about the cities that have features like the home town of the tourist, so we check what cluster, if any, was assigned to the traveler's city:


```
df_reduced['cluster'].unique()

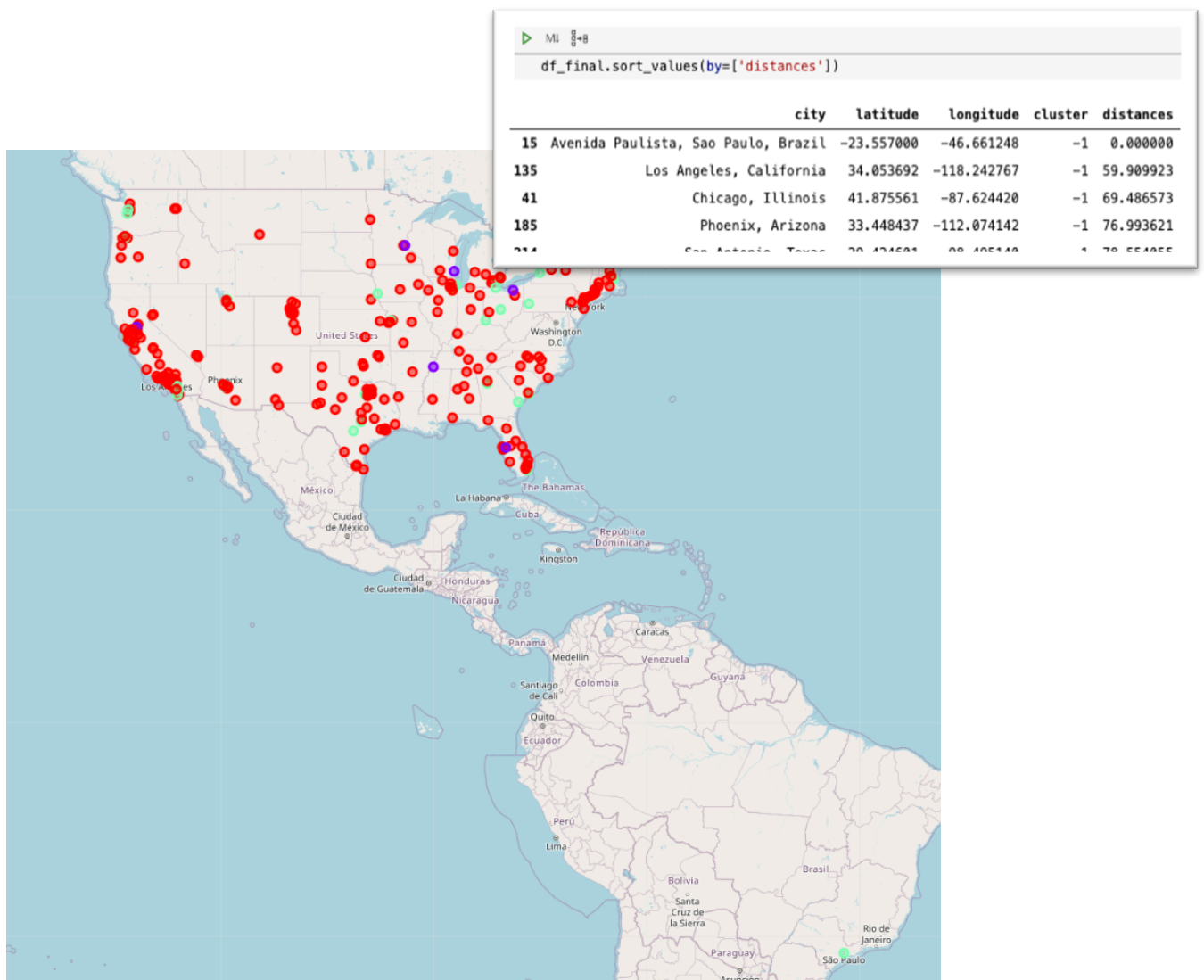
array([ 0, -1,  1])

df_reduced: DataFrame
df_reduced[df_reduced['city'] == home]
```

	city	Comp_1	Comp_2	Comp_3	Comp_4	Comp_5	Comp_6	Comp_7	Comp_8	Comp_9	Comp_10	Comp_11	cluster
15	Avenida Paulista, Sao Paulo, Brazil	85.057799	-17.772735	-7.875107	0.970029	0.935629	-0.993869	0.112833	-0.552308	0.021891	0.635079	0.117906	-1

It was assigned cluster “-1” which is code for “noise”, meaning the city didn’t show features that associate it with cities of the 2 clusters identified by the model.

One thing we can do is to calculate the distance between our “detached” data point corresponding to Sao Paulo, and see what would be the closest city in this multidimensional space using the Euclidian distance. What we see is that Los Angeles seem to be the closest option to Sao Paulo, and we must remember that this might mean very little since Los Angeles also had no clear connection with a major cluster detected by the model.



Some adjustments can be made to the model. New features like weather over a period or more data points could enhance the model making it easier to see connections and create clusters.

This is not a problem where a ground truth is known, so no metric of error can take place. The question of finding some structure in the data and having insights about how the data is connected had its solution with the definition of 2 clear clusters and a few cities that don't quite fit these 2 profiles. Perhaps adding cities from other countries we have the diversity necessary to fill in the gaps.