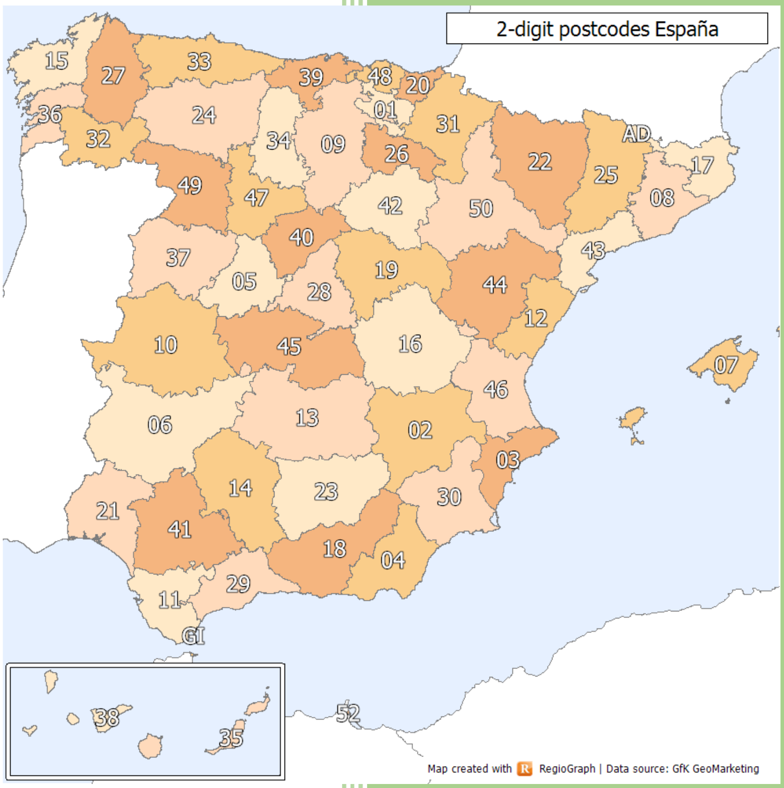
2021

Spanish postal code content-based recommendation system



Adrián Rejas Conde

IBM Data Science certificate capstone project

6-6-2021

Contenido

1. Introduction

When people need to move into another city, choosing the right area to live in the new city, without knowing the new city, could be a challenge. There are many variables in choosing the right area to live, and the knowledge about the new city may be limited, which can result in choosing an area which will not be the suitable for a person, deriving in a wide range of problems, from logistics problems like increases in travelling times to the different places and venues visited in the day-to-day to psychological problems derived from unhappiness with the place chosen.

Although each country, or even each city in the same country has different ways to organize the neighbourhoods and areas of a city, there is one which can be considered universal, which is the postal code area.

My idea for this project is to use a set of data science techniques for designing a content-based recommendation system for choosing the postal code area to live in a new city which will fit best with the user based on information provided by him.

The user will be able to choose whether he wants a recommendation based on the similarity with the neighbourhood or the density of venues. If the similarity approach is taken, the following data will be required to the user:

- The postal code where he is living right now.
- The city where he is planning to move in.
- The type of amenities he is interested in.

If the density approach is taken, the following data will be required to the user:

- The city where he is planning to move in.
- The type of amenities he is interested in.

For the first approach, the idea will be set in the 15 most populous cities of Spain, in order to make a first prototype with enough complexity.

The target users for the recommendation systems will have the following characteristics:

- People living in Spain or thinking about moving into Spain.
- People thinking about moving into another home, in the same city or in another Spanish city.
- People who consider the amenities of the area as an important feature for choosing a home.

2. Data to be used

For developing the idea proposed, both data about postal code areas of Spain and the venues located at each of the postal code areas is required.

For data regarding Spanish postal codes, the information provided by the Spanish Mail Delivery Office will be used. This information provides data about each postal code of Spain, the city or town it belongs to and its geographical boundaries. The information to be requested to this data source is a list of postal codes, each of one with the following fields:

- ID of the postal code area.
- City where the postal code area is located.
- Geographical boundaries of the postal code area in the form of GeoJSON file.

Thanks to Inigo Flores, this information is available in CSV and GeoJSON formats at the following URL:

https://github.com/inigoflores/ds-codigos-postales

As commented in the introduction, the Spanish Mail Delivery Office dataset will be filtered for getting the postal codes corresponding to the 15 most populous cities of Spain. These 15 cities will be configured by using a handmade JSON file.

For data regarding venues located at each of the postal code areas of Spain, Foursquare database will be used. Foursquare provides a REST API for connecting with Foursquare database, providing the required methods for getting basic information about venues located in an area. The information will be requested to Foursquare API for each postal code area of the ones selected previously, getting latitude, longitude and area radius from the geographical boundaries of each of the postal code areas. For each postal code area, the following information will be requested:

- Number of venues in the area.
- List of the venues of the area, with the following info for each of them:
    - Venue name.
    - Venue location (latitude and longitude).
    - Venue category (both primary, secondary and final category).

3. Methodology

In this section the methodology for creating the recommendation system will be shown. It will consist of the following steps:

1. Preparation of the development environment.
2. Data obtention.
3. Data processing.
4. Build and tune recommendation system algorithm.
5. Build visualization tools.
6. Build application.

3.1. Development environment

As development environment, a Python Jupyter notebook run over Anaconda Python distribution for Windows has been chosen.

The use of a Jupyter notebook was imposed by the requirements of the project. But its ease of use, the availability of a wide range of environments for running it (both local and cloud-based solutions), its portability to another development environments based also on Jupyter notebooks makes Jupyter notebooks and its combination between documentation and code makes them a perfect tool for this project.

As programming language, Python has been chosen, because of its portability, its ease to use and the wide range of libraries available which are really useful for data science. Between the libraries used in the project, these are the most important:

- Pandas: for the management of datasets.
- Numpy: for the management of lists.
- Scipy: for statistical analysis and the calculation of Euclidean distances matrixes, required for the recommendation system (seen later).
- Geocoder: for getting coordinates from addresses.
- Folium: for the rendering of maps.
- Matplotlib: for the rendering of graphs.
- iPython: for the development of applications with user interface.

As running environment, several cloud-based alternatives have been tested, like Watson Studio and LabsCognitive. However, they have given problems when rendering user interfaces developed with iPython and other similar libraries. Because of this, a decision has been made for using a local running environment, choosing Anaconda Python distribution for Windows because it is the most widely used Python environment for Windows.

The first code steps of the Jupyter notebook have been the download and importation of the required Python libraries and the implementation of a set of functions for working with coordinates.

3.2. Data obtention

Once prepared the development environment, it is time for getting the data required for creating the recommendation system. Three types of data are required for the building of the recommendations system:

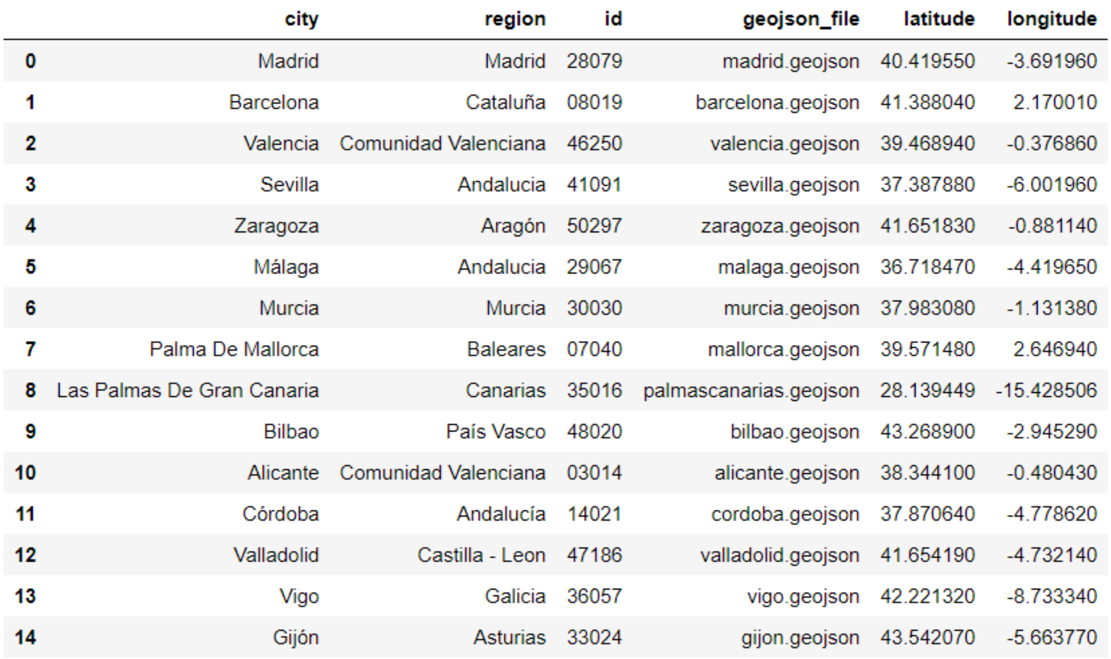
- Data about cities of Spain.
- Data about postal codes areas located at these cities.
- Data about venues located at these postal code areas.

3.2.1. Cities Data

The first step will be to define a pandas dataframe with the cities which will be included in the recommendation system. The 15 most populous cities of Spain have been selected, defining city, region, ID of the city (got from an observation of the postal code dataset) and name of the file with the GeoJSON information about the postal codes of each city.

The list of cities and the info provided has been created manually, because of the few info for each city, the location will be got using ArcGIS geocoder library.

The dataset obtained has the following format:



	city	region	id	geojson_file	latitude	longitude
0	Madrid	Madrid	28079	madrid.geojson	40.419550	-3.691960
1	Barcelona	Cataluña	08019	barcelona.geojson	41.388040	2.170010
2	Valencia	Comunidad Valenciana	46250	valencia.geojson	39.468940	-0.376860
3	Sevilla	Andalucia	41091	sevilla.geojson	37.387880	-6.001960
4	Zaragoza	Aragón	50297	zaragoza.geojson	41.651830	-0.881140
5	Málaga	Andalucia	29067	malaga.geojson	36.718470	-4.419650
6	Murcia	Murcia	30030	murcia.geojson	37.983080	-1.131380
7	Palma De Mallorca	Baleares	07040	mallorca.geojson	39.571480	2.646940
8	Las Palmas De Gran Canaria	Canarias	35016	palmascanarias.geojson	28.139449	-15.428506
9	Bilbao	País Vasco	48020	bilbao.geojson	43.268900	-2.945290
10	Alicante	Comunidad Valenciana	03014	alicante.geojson	38.344100	-0.480430
11	Córdoba	Andalucía	14021	cordoba.geojson	37.870640	-4.778620
12	Valladolid	Castilla - Leon	47186	valladolid.geojson	41.654190	-4.732140
13	Vigo	Galicia	36057	vigo.geojson	42.221320	-8.733340
14	Gijón	Asturias	33024	gijon.geojson	43.542070	-5.663770

Figure 1. Cities dataset format

It provides the following info:

- Name of the city.
- Region where the city is located.
- Id of the city, according to the Spanish Postal Service.
- Name of the GeoJSON file defining the postal code areas in the region where it is located.
- Coordinates of the city centre.

At the following figure it can be seen a map representing the cities used for the recommendation system:

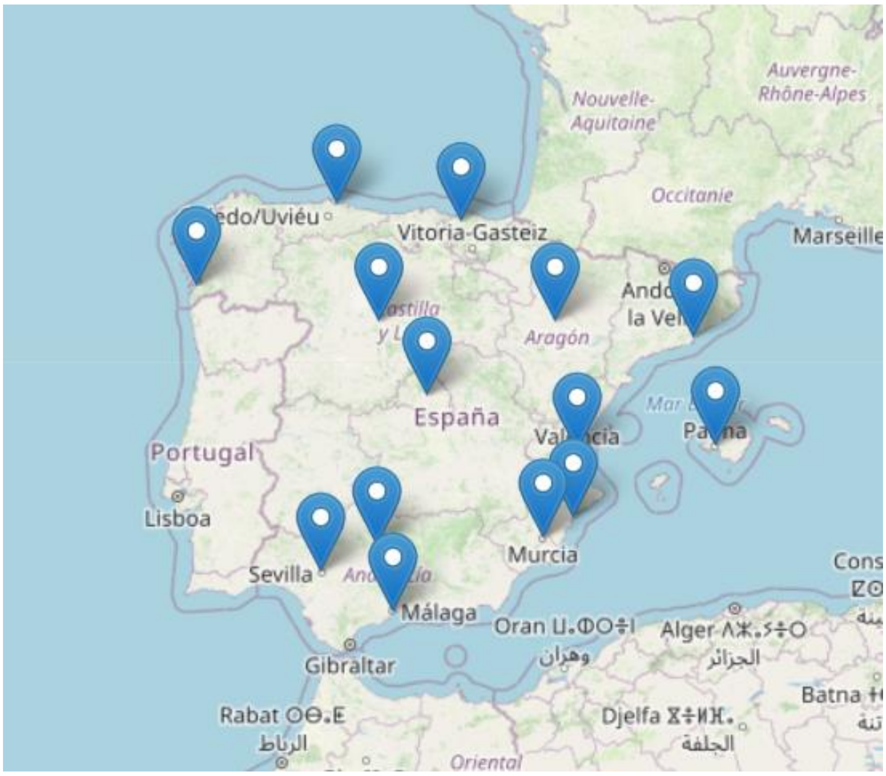


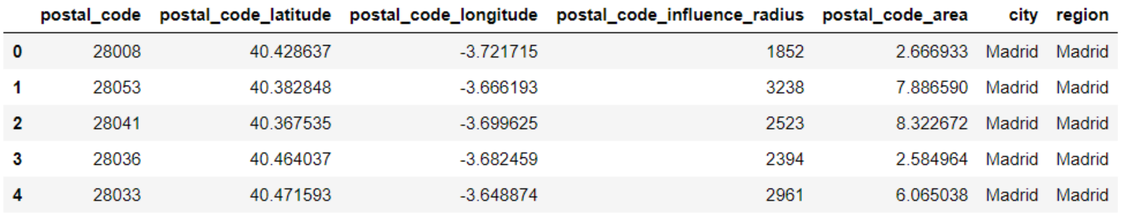
Figure 2. Map with used for the recommendation system

3.2.2. Postal codes data

The next step is to get the data of the postal codes, that is the name, city and region belonging to, latitude and longitude location and influence radius. For getting this data in the form of a dataset, the following process has been done:

- Get from https://github.com/inigoflores/ds-codigos-postales The CVS with the info about the postal codes in Spain and the GeoJSON files with the postal codes delimitations of all the regions of the cities to be used by the recommendation system.
- Filter the postal codes dataset by excluding those postal codes whose city ID does not correspond to a city ID of the cities dataset previously got.
- Filter the GeoJSON files for creating GeoJSON files corresponding to the cities to be used, by removing from each of the original postal codes those areas whose postal code ID is not included in the list of postal codes of the city in question. The filtered GeoJSON files will be saved as different files from the previous ones.
- Using the filtered info about the postal code's info, the ArcGIS Geocoder library and the GeoJSON files, work out location, latitude, area in square kilometres and influence radius in meters of each postal code area.
- Gather all info about postal codes in a pandas dataframe.

The resulting dataframe consisted of 435 entries, and it has this format:



	postal_code	postal_code_latitude	postal_code_longitude	postal_code_influence_radius	postal_code_area	city	region
0	28008	40.428637	-3.721715	1852	2.666933	Madrid	Madrid
1	28053	40.382848	-3.666193	3238	7.886590	Madrid	Madrid
2	28041	40.367535	-3.699625	2523	8.322672	Madrid	Madrid
3	28036	40.464037	-3.682459	2394	2.584964	Madrid	Madrid
4	28033	40.471593	-3.648874	2961	6.065038	Madrid	Madrid

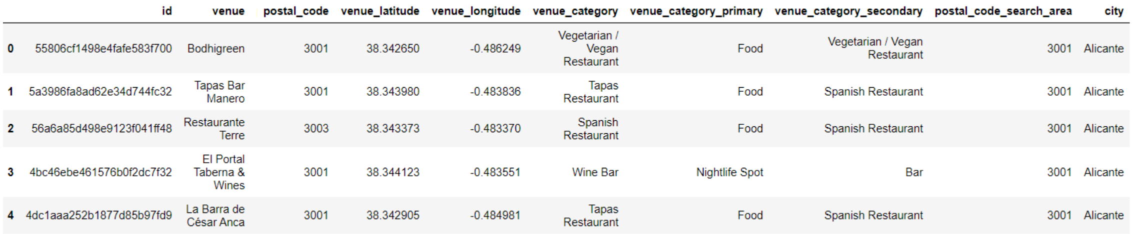
Figure 3. Postal codes dataset format

3.2.3. Venues dataset

The last dataset to get is the list of most relevant venues of each postal code to be analysed. In order to do so, Foursquare places API will be used. The process will be the following:

- The credentials for the Foursquare API will be set for being used by future calls.
- Foursquare contemplates three possible levels of categories: primary, secondary and final. The whole category tree will be requested to Foursquare API, and they will be defined the functions for getting both primary and secondary categories from a final category.
- A function will be defined for getting from the info about a postal code a list of all the venues of the area, using paginated Foursquare API call to the explore endpoint. As input variables to the function, they will be used the longitude and latitude location and the influence radius for defining the searching area, so as the city, region and postal code ID. As a result, it will be obtained the total number of venues in the area and a list of venues. For each venue, it will be provided the ID and name of the venue, it is location, the categories corresponding to the venue and address info.
- Another function will be defined for getting the combined list of all the venues of all postal codes inside a postal code dataset, getting a dataframe will all venues info explained previously and a dataframe with the total number of venues per postal code as a result.

The resulting dataframe consisted of over 42.000 entries, and it has this format:



	id	venue	postal_code	venue_latitude	venue_longitude	venue_category	venue_category_primary	venue_category_secondary	postal_code_search_area	city
0	55806cf1498e4fafe583f700	Bodhigreen	3001	38.342650	-0.486249	Vegetarian / Vegan Restaurant	Food	Vegetarian / Vegan Restaurant	3001	Alicante
1	5a3986fa8ad62e34d744fc32	Tapas Bar Manero	3001	38.343980	-0.483836	Tapas Restaurant	Food	Spanish Restaurant	3001	Alicante
2	56a6a85d498e9123f041ff48	Restaurante Terre	3003	38.343373	-0.483370	Spanish Restaurant	Food	Spanish Restaurant	3001	Alicante
3	4bc46ebe461576b0f2dc7f32	El Portal Taberna & Wines	3001	38.344123	-0.483551	Wine Bar	Nightlife Spot	Bar	3001	Alicante
4	4dc1aaa252b1877d85b97fd9	La Barra de César Anca	3001	38.342905	-0.484981	Tapas Restaurant	Food	Spanish Restaurant	3001	Alicante

Figure 4. Venues dataset format

3.3. Data processing

Once obtained the data, it needs to be processed in order to create the content-based recommendation system.

The goal of this section is to get a dataset of the postal codes and data for each postal code which will serve for comparing postal code areas between them.

The approach chosen has been to define for each postal code area the density of venues per square kilometre of each of the venue category types available. Other possibilities considered were to get the percentage of venues of each type inside a postal code area and the number of venues at each postal code area. The percentage of venues was discarded because it gave info about the distribution of venue types, but not about the number of venues. And the raw number of venues was not aware of the differences in sizes between different postal codes. So, the choice done was use the density of venues.

For the recommendation system, the secondary category will be use, the first category is too generic, while the final category may be too detailed.

Before any other processing, it is mandatory to filter the venues dataset with the following conditions:

- It cannot exist two repeated tuples of venue ID and postal code.
- It cannot exist inputs with a postal code not considered in the postal code dataset.

Once done, the venues dataset pass from over 42.000 inputs to over 21.000, half of the original dataset.

Once filtered, the venues dataset will be used for creating a pandas dataframe with the info for each of the postal codes extended with the density of each of the secondary categories in every postal code per square kilometre. The process will be the following:

- Get a dataframe with one-hot encoding of the secondary category and the postal code for each of the venues.
- Enter an all-zeroes entrance if a postal code has no venues.
- Get a dataframe with the number of venues of each category in every postal code.
- Merge the info of the newly created dataframe with the postal codes info and postal codes total venues dataframes.
- For each of the postal codes and venue categories, divide the number of venues by the area in square kilometres in order to get the number of venues per square kilometre of each category for each postal code.
- Drop redundant info found in other datasets from the newly created venues density dataframe.

3.4. Build recommendation algorithm

Once processed the data, it needs to be processed in order to create the content-based recommendation algorithm. The recommendation system will be created in the form of a function. The input of the function will be the following:

- criterium: 'similarity' for a recommendation based on similarity with a postal code, 'density' for a recommendation based on the density of venues.
- dataframe: Dataframe with the venues density of the postal codes to be considered by the recommendation system.
- similarity_reference: Maximum distance possible between two postal codes. Used only with similarity criterium in order to use it for working out the similarity between postal codes.
- reference: if similarity criterium used, venues density of the postal code to be taken as the reference for the similarity.

- venues: If set, considering for the recommendation system just the venue categories listed here. If not set or empty, consider all venue categories.
- Recommendator_enhancer: the recommendation mark, when normalized to 1 and before converted to percentage, will be raised to the power of this parameter in order to calibrate the differences at high levels, enhancing the difference with a value >1 or decreasing the difference with a value <1.
- Similarity_enhancer: if similarity criterium used, the similarity mark, when normalized to 1 and before converted to percentage, will be raised to the power of this parameter in order to calibrate the differences at high levels, enhancing the difference with a value >1 or decreasing the difference with a value <1.

The output will be tuple with two arrays of the same length:

- The first one will provide a recommendation mark for each of the postal codes provided in 'dataframe', being the most recommended postal code marked with 100% and the least recommended postal code with 0%.
- The second one will depend on the criterium chosen:
    - If similarity chosen, it will provide a mark of the similarity of each of the postal codes provided in 'dataframe' with the postal code provided in 'reference'.
    - If density chosen, it will provide a combined density of all venue categories considered for each of the postal codes provided in 'dataframe'.

Once implemented the recommendation algorithm will need to be tested with all possibilities of reference postal codes and cities to move in, so as a selection of values for the hyperparameters of the algorithm, which are the enhancer parameters.

The enhancer parameters can take a value bigger than 0, with a value between 0 and 1 incrementing the lowest values and a value over 1 decrementing the lowest values, so the algorithm will be tested with both kind of values.

From the testing algorithm, the following info will be stored in the way to select the most suitable values for the hyperparameters:

- Top three recommendation marks.
- Number of recommendation marks over 75%, over 50% and over 25%.
- If similarity criterium, top three similarity marks.
- f similarity criterium, number of similarity marks over 75%, over 50% and over 25%.

For tuning the hyperparameters, compare the results obtained in order to see which one is giving a more distributed marks, either recommendation or similarity. The choice will be based on the comparison between the histograms of the info previously gathered for the different enhancer possible values.

One thing to note is respecting the venue categories selected. As testing all venue category combinations for all postal codes for all cities will last a lot of time, and the most of times the algorithm will be used with no venue category selection set (all venue categories considered), that is the way the algorithm will be tested.

3.4.1. Recommendation for similarity criterium

The algorithm for the similarity criterium will be a custom version of the Nearest Neighbours algorithm. The reason behind using a custom implementation instead of the one provided by SKLearn Python library is that this library implementation provides as output the distances ordered from lowest to highest, not in the same order as in the input. Because of this and since it is a simple algorithm to be implemented by using Scipy library, it has been considered a better option a custom implementation. The distances will be calculated by using the Euclidean function.

The recommendation mark will be worked out by calculating the Euclidean distance between the postal codes (taking the venue categories density as parameters) in the form of array, normalizing the value to the max value, inverting the result (1 minus self) and transforming the value to percentage. Before transformed to percentage, it will be raised to the power of a number set by the 'recommendation_enhancer' parameter in order to calibrate the differences in recommendations between postal codes.

The similarity mark will be worked out in the same way, but the normalization will be used by using ‘similarity_reference’ parameter (which will represent the maximum distance possible between two entries) and the calibration will be done by using the ‘similarity enhancer’.

Once implemented and tested, the results got depending on the recommendation enhancer are seen on the next page. From there, the following conclusions can be extracted:

- All recommendation enhancers provide a recommendation mark of 100% for the best area, confirming the good working of the recommendation algorithm according to the description done.
- The top 3 areas recommended are generally giving a high value (>90) with all enhancer values tested, being this value inversely proportional to the enhancer value.
- The number of areas recommended with >75% of recommendation rate is smaller with bigger enhancer values.

As the target is to give a clear recommendation, with all options available to be checked but few options with high recommendation marks given (5 or so if possible), but still with some differences variety in the recommendation marks, **the recommendation enhancer value chosen for the similarity criterium will be 8**.

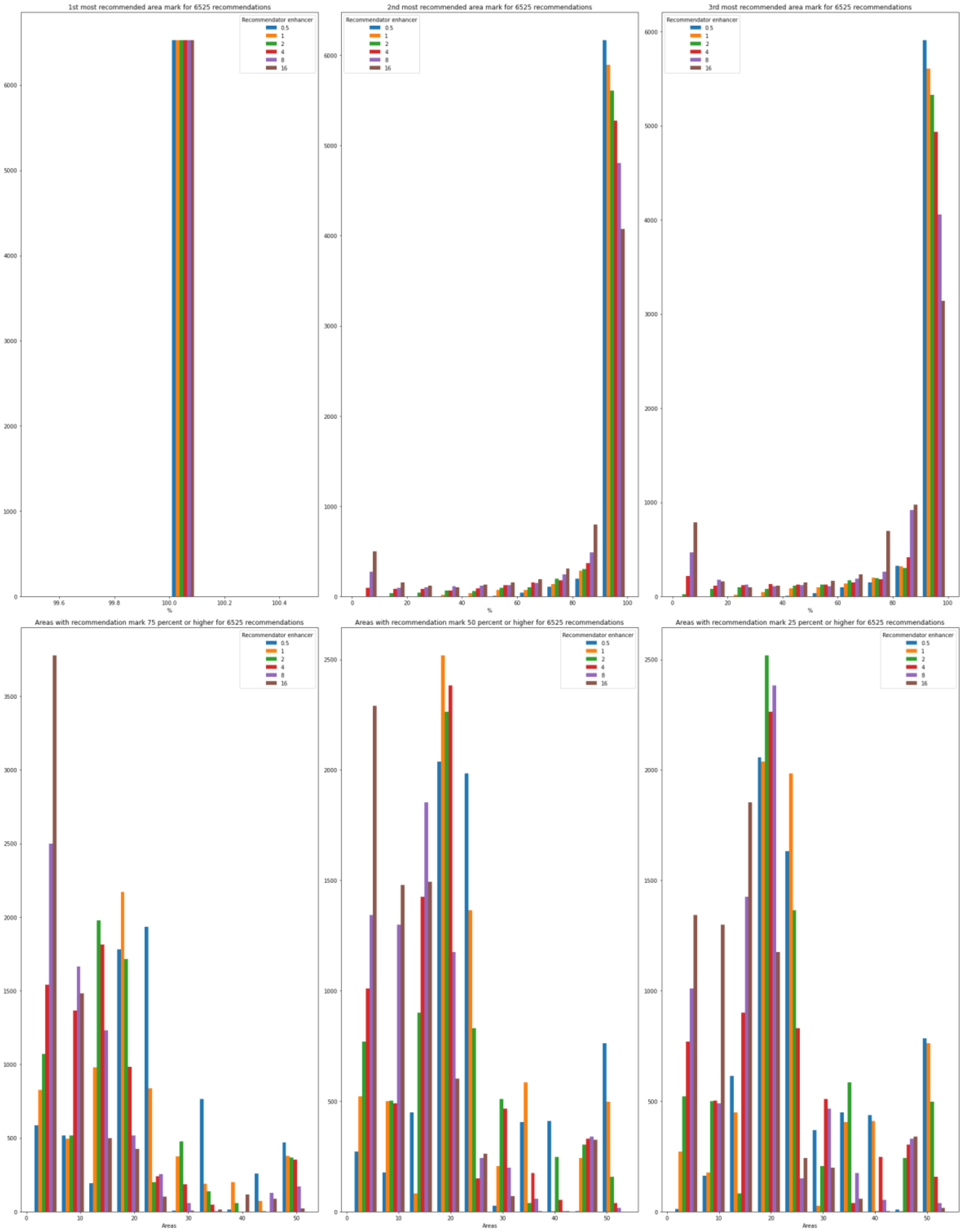


Figure 5. Test results of recommendation enhancer with similarity criterium

And finally, the results got depending on the similarity enhancer are seen on the next page.

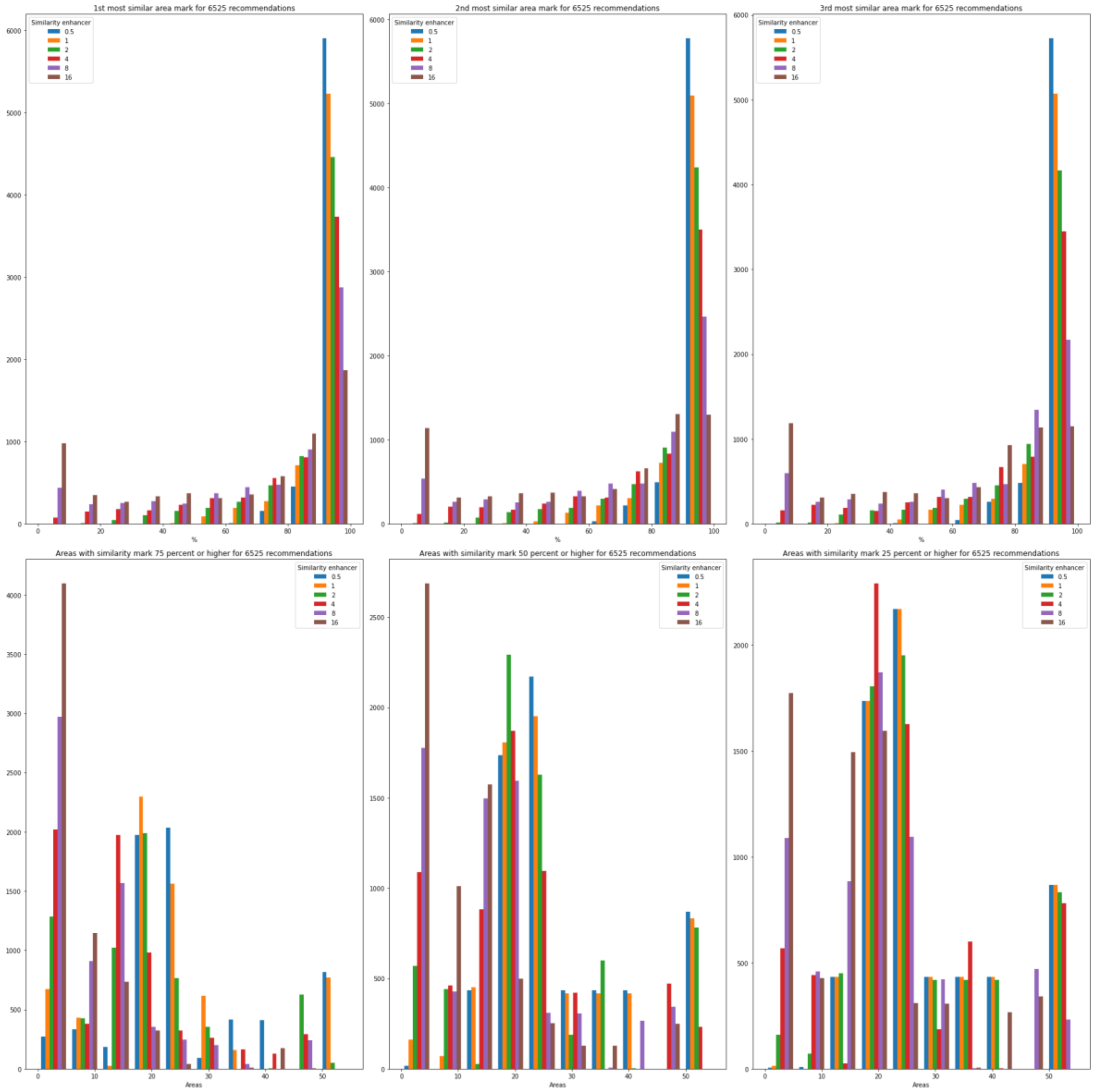


Figure 6. Test results of similarity enhancer with similarity criterium

From there, the following conclusions can be extracted:

- The top 3 areas recommended are giving a recommendation mark inversely proportional to the enhancer value. High enhancer values provide a lot of top similarity scores with small similarity marks, while small enhancer values provide small top marks.
- The number of areas recommended with >75% of recommendation rate is smaller with bigger enhancer values, with very few areas with high similarity mark with high values, while with low enhancer values it is common to find a lot of areas with high similarity mark.

As the similarity value is strictly informative, it does not affect the recommendation mark and it is assumed all postal codes are similar between them up to a point (they all are postal code areas of relatively big cities of Spain), but a small enhancer value makes the similarity marks too high, it is recommendable to choose an average value. Considering this, **the similarity enhancer value chosen for the similarity criterium will be 4**.

3.4.2. Recommendation for density criterium

In the case of density criterium, the recommendation algorithm implementation will be easier.

The recommendation mark will be worked out by getting an array of the density for each of the postal codes analysed, normalizing the value taking using the max value and transforming the value to percentage. Before transformed to percentage, it will be raised to the power of a number set by the 'recommendation_enhancer' parameter in order to calibrate the differences in recommendations between postal codes.

Once implemented and tested, the results got depending on the density enhancer are seen on the next page. The following conclusions can be extracted:

- All recommendation enhancers provide a recommendation mark of 100% for the best area, confirming the good working of the recommendation algorithm according to the description done.
- The second and third highest recommendation marks used to be small with higher values like 8 and 16.
- The number of areas recommended with >75% of recommendation rate is smaller with bigger enhancer values.

As the target is to give a clear recommendation, with few options available to be checked but with at least some of them with a higher value, it is better to take a small value based on the data shown. Based on this, **the recommendation enhancer value chosen for the similarity criterium will be 1/2**, because that value will give between 2 and 5 recommendations over 75% most of times.

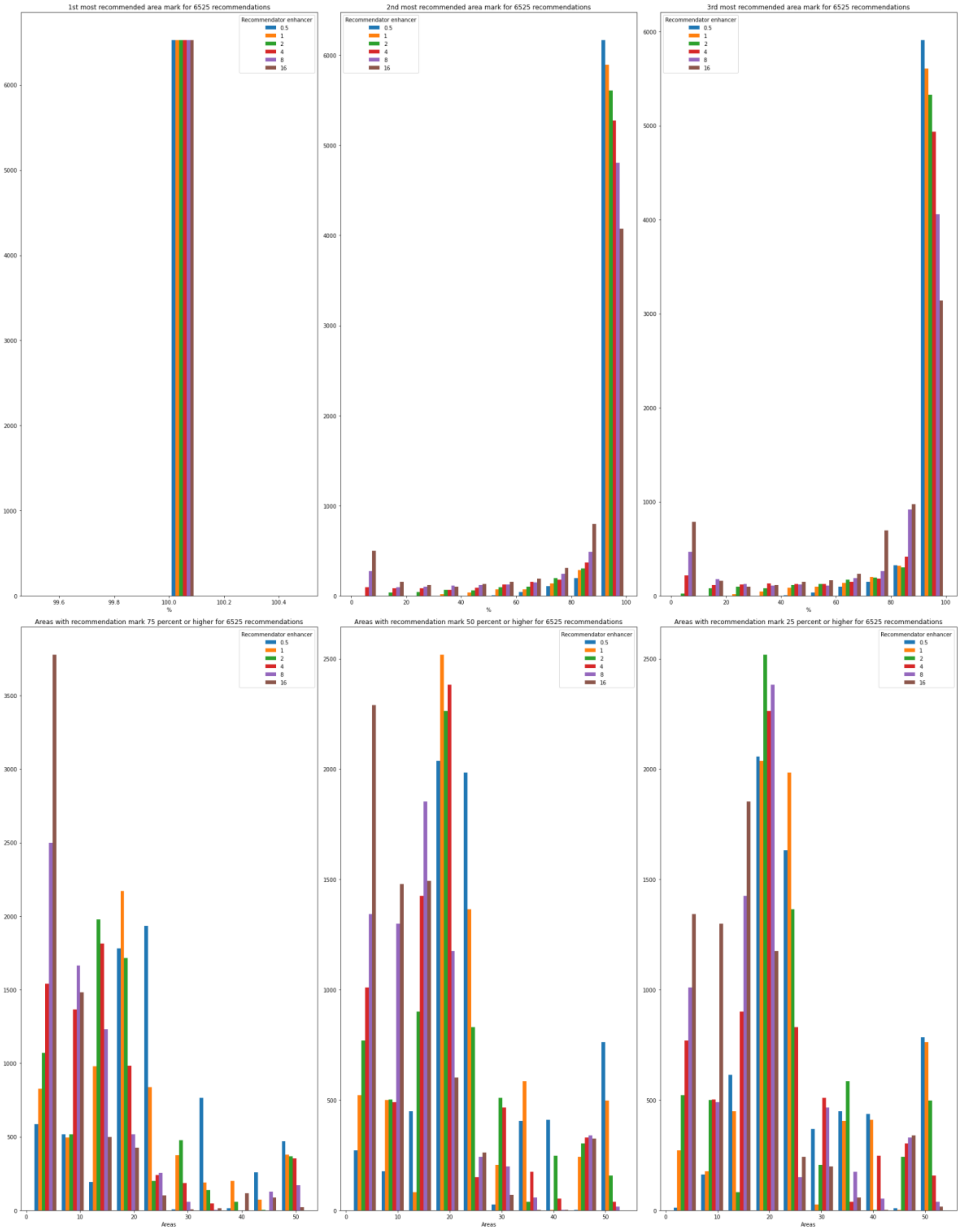


Figure 7. Test results of recommendation enhancer with density criterium

3.5. Build visualization tools

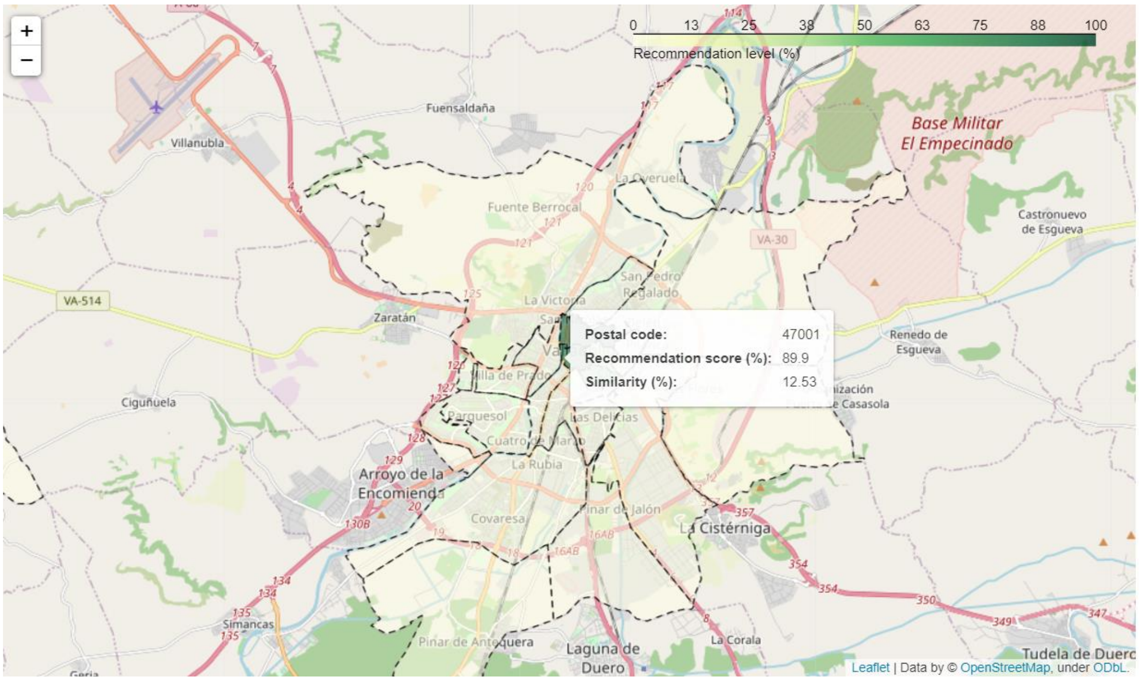
Once build the function implementing the recommendation algorithm, a visualization tool needs to be provided in order to get a recommendation from a few inputs.

As visualization tool, two functions will be implemented.

- Both functions will show a choropleth map of a city chosen to move in, remarking the postal code areas of the city with a green colour scale, being the darkest postal code areas the most recommended ones for living considering the criterium chosen, the postal code chosen as reference and the venue categories the user is interested in.
- The maps generated by both functions will show tooltips when the mouse is placed over the area, showing the postal code ID, the recommendation mark for this postal code and the similarity mark (in case of similarity criterium) or the density of venues (in case of density venue).

The first function will be used in case of similarity criterium chosen, and its inputs are:

- postal_code: postal code used as reference (normally the one where the user is currently living).
- city_to_movein: city to be shown on the map and whose postal codes will be given a similarity and a recommendation mark (normally the city where the user wants to move in).
- venues_selected: Venue categories that will be considered for creating the similarity and the recommendation marks (normally the venue categories the user is interested in).

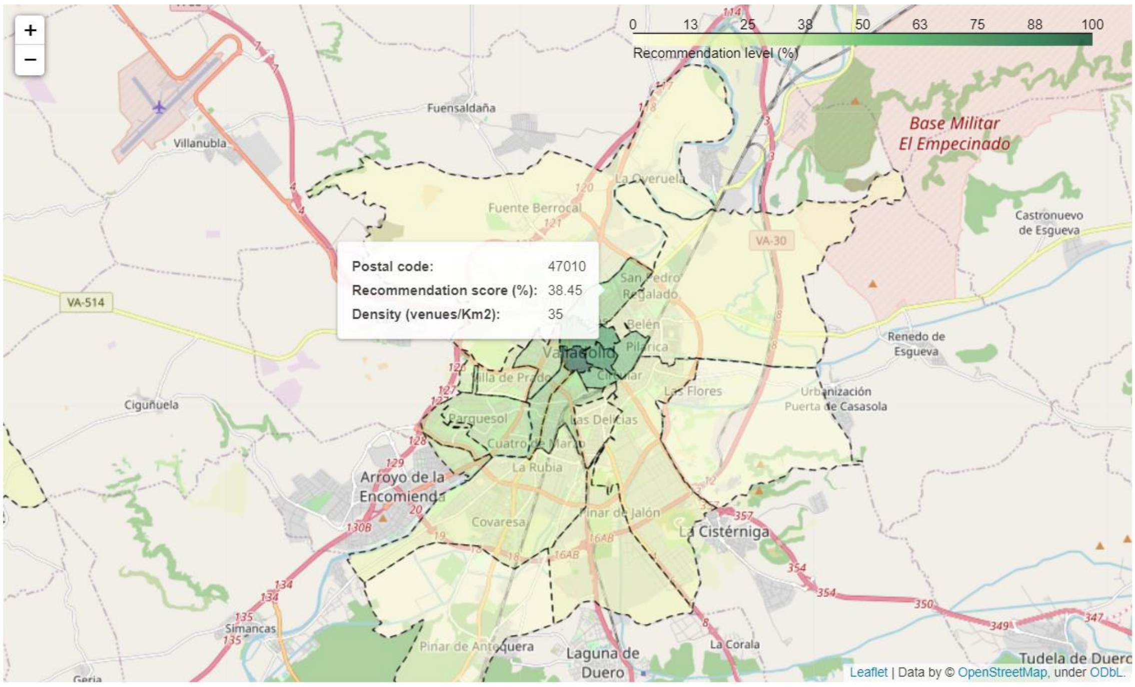


Postal code: 47001
Recommendation score (%): 89.9
Similarity (%): 12.53

Figure 8. Visualization tool output for similarity criterium

The second function will be used in case of density criterium chosen, and its inputs are:

- city_to_movein: city to be shown on the map and whose postal codes will be given the density of venues and a recommendation mark (normally the city where the user wants to move in).
- venues_selected: Venue categories that will be considered for getting the density of venues and creating the recommendation marks (normally the venue categories the user is interested in).



Postal code:	47010
Recommendation score (%):	38.45
Density (venues/Km2):	35

Figure 9. Visualization tool output for density criterium

Both functions have been implemented using Folium Python library, it is capacity to process GeoJSON files in order to create choropleth maps and using the recommendation algorithm implemented previously with the hyperparameters worked out previously for a fine tuning.

3.6. Build application

Finally, with the recommendation algorithm and visualizer functions yet implemented, it is time for creating an application using those functions in order to create a simple interface the user can interact with in order to get the recommendations desired.

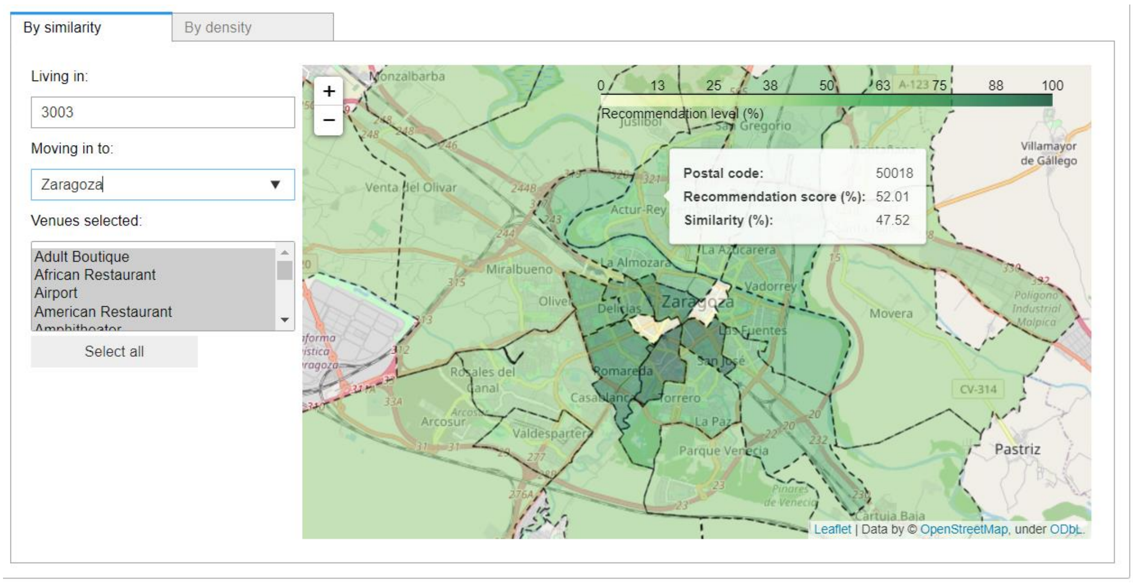
For creating the application, Ipython module classes will be used, allowing to create an application the user can interact with. The callbacks used for managing user's interaction will make use of the visualization functions implemented previously, which at the same time will make use of the recommendation algorithm implemented previously. The inputs of the application will be in the following way:

- The criterium used will be chosen using two tabs, one for similarity and the other one for density.
- The city to move in and, if similarity criterium chosen, the reference postal code will be chosen by using dropdown selectors, with the possibility to write requested values in order to find them easily.

- The venue categories to use in the recommendation will be chosen by using a multiselector list, will all categories selected by default.

This is a prototype of the final application, still pending to be enhanced visually and with a more robust error control. However, the application is fully functional, with a basic but effective style and a basic error control but able to handle all problems in a basic way.

The application, although with a basic UI, works well, providing a recommendation about postal codes where to live based on information provided by the user.



By similarity | By density

Living in: 3003

Moving in to: Zaragoza

Venues selected:
Adult Boutique
African Restaurant
Airport
American Restaurant
Amphitheater

Select all

Postal code: 50018
Recommendation score (%): 52.01
Similarity (%): 47.52

Figure 10. Application resultant from the project

4. Results

Once build and tuned the function implementing the recommendation algorithm, the visualization tools and the application to use both, it is time to analyse the results obtained with the recommendation system.

Though an exploratory analysis done by using the application, there are some hypotheses which according to results observed with manual testing are considered as worth to be analysed. These are the following:

- When using the similarity criterium, there is a correlation between the relative distance of the most recommended postal code area from its city centre and the relative distance of the postal code area taken as reference from its city centre.
- When using the similarity criterium, there is a correlation between the venue type which is more common in the most recommended postal code area from its city centre and the venue type which is more common in the postal code area taken as reference.
- When using the similarity criterium, the two previous correlations, if existing, can vary depending on the type of venues considered for doing the recommendation.
- When using the density criterium, there is a correlation between the density of venues of a postal code and the distance of the postal code from its city centre.

- When using the density criterium, the correlation, if existing, between the density of venues of a postal code and the distance of the postal code from its city centre can vary depending on the type of venues considered for doing the recommendation.

For confirming the hypothesis regarding the similarity criterium, the recommendation algorithm will be used for doing a series of recommendations. The recommendations will be done using the following categories to be considered:

- One recommendation containing all the venue categories available.
- Several recommendations, each one containing all the venue categories available with one of the nine primary categories in common.

4.1. Results for similarity criterium

For each combination of postal code area available and city available to move in, the recommendation algorithm will be used for getting a series of recommendations based on similarity, one for each of the venue recommendations lists defined to be considered. For each of these recommendations, the following data will be stored in a dataframe:

- User hypothetical input (reference postal code, city to move in and venue categories to consider).
- Distance of the reference postal code from its city centre (normalized to the influence radius of the city).
- Distance of the most recommended postal code from its city centre (normalized to the influence radius of the city).
- Both first and second venue category with the highest density in the reference postal code.
- Both first and second venue category with the highest density in the most referenced postal code.

For checking the hypothesis of the correlation between the distances from the city centre of the postal code taken as reference and the most recommended postal code, the following info will be show for each of the venue category lists used previously:

- A scatter plot showing the relationship of both distances for all tests done with these venue category lists to be considered.
- The Pearson coefficient and P-value of both distances for each of these venue category lists to be considered.

The results can be seen in the following page.

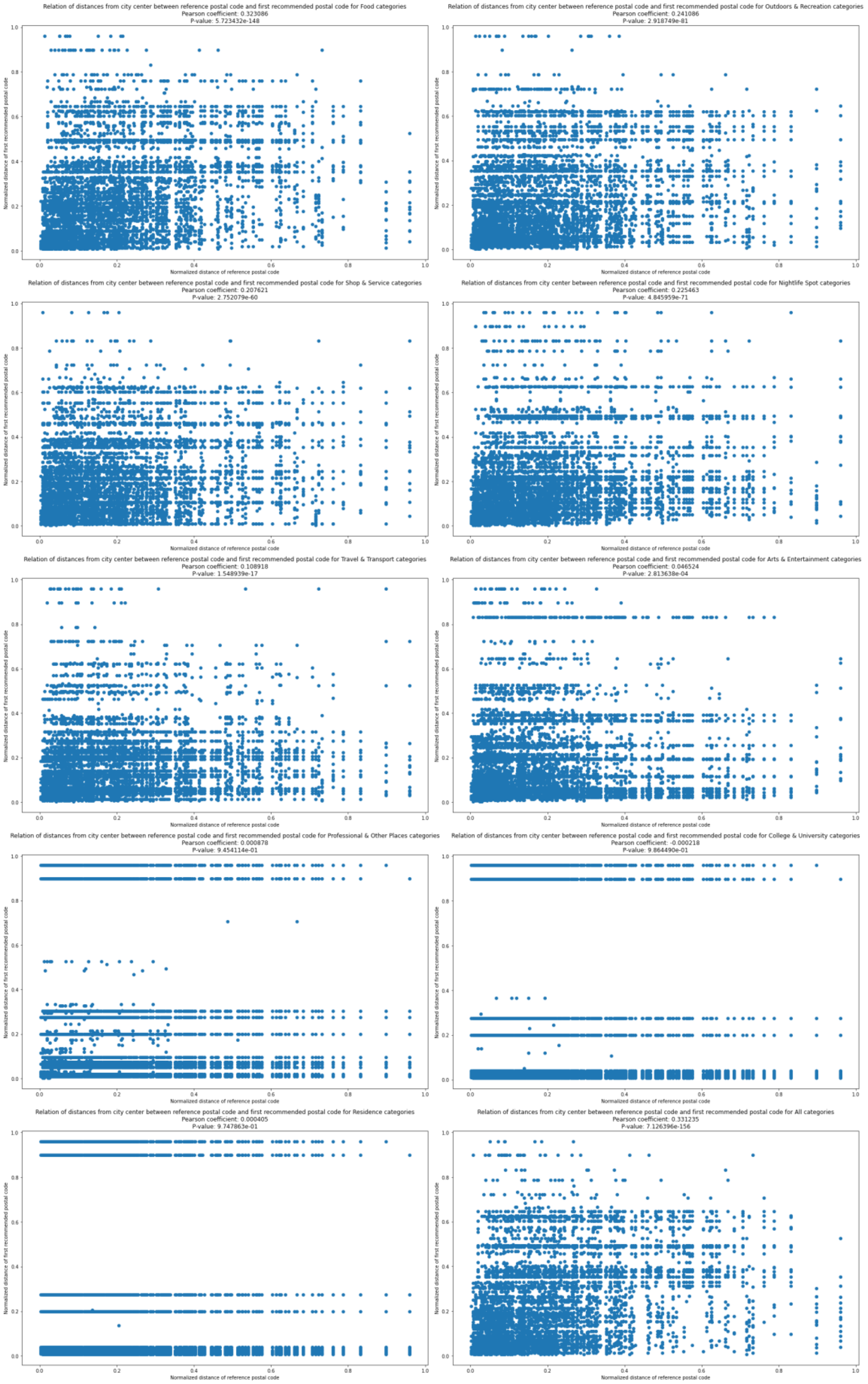


Figure 11. Correlation between distances to centre for similarity criterium

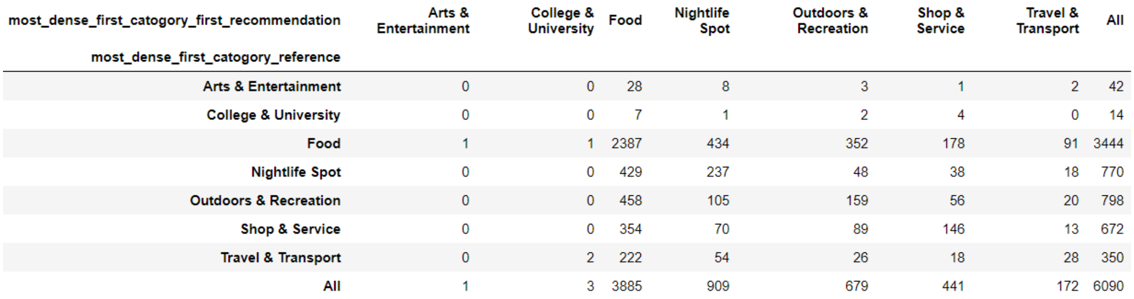
The scatter plots results show at first to be very disperse, but showing some trends confirmed by the Pearson correlation values, which are the following:

- There is a positive moderate correlation between distances when using all categories and food related categories.
- There is a positive weak correlation between distances when using nightlife, outdoors, shopping and transport related categories.
- There is a positive extremely weak correlation between distances when using arts & entertainment related categories. This correlation is non existent according to widely accepted Pearson correlation levels generally accepted, but because of it is extremely low P-value it can be considered as existent but extremely weak.
- There is no existing correlation between distances when using professional, college and residence categories.

But's it is necessary to remark that the P-value at all cases tend to be extremely close to 0 in all cases with some correlation, denying the null hypothesis.

So, it can be considered that there is a moderate to week but significant positive correlation between the distances from the city centre of the reference postal code and the most recommended postal code if using food, nightlife, outdoors, shopping, transport and arts & entertainment related categories, and using all categories.

For checking the dependency between the most common venue category between the reference postal code and the most recommended postal code, a Chi square test will be built in order to compare both variables.



most_dense_first_catogory_first_recommendation	Arts & Entertainment	College & University	Food	Nightlife Spot	Outdoors & Recreation	Shop & Service	Travel & Transport	All
most_dense_first_catogory_reference								
Arts & Entertainment	0	0	28	8	3	1	2	42
College & University	0	0	7	1	2	4	0	14
Food	1	1	2387	434	352	178	91	3444
Nightlife Spot	0	0	429	237	48	38	18	770
Outdoors & Recreation	0	0	458	105	159	56	20	798
Shop & Service	0	0	354	70	89	146	13	672
Travel & Transport	0	2	222	54	26	18	28	350
All	1	3	3885	909	679	441	172	6090

Figure 12. Correlation between most common venues for similarity criterium

According to the results of the Chi square test, no correlation seems to exist between the most common venue category of the reference postal code and the recommended one. The most common venue category at each recommended postal code is usually a child of Food first category, no matter the most common venue category at the reference postal code.

This points out to the possibility of Food categories having an excess of influence over the results. In order to check this, a histogram about the type of venues presented in the venues dataset used for working out the densities.

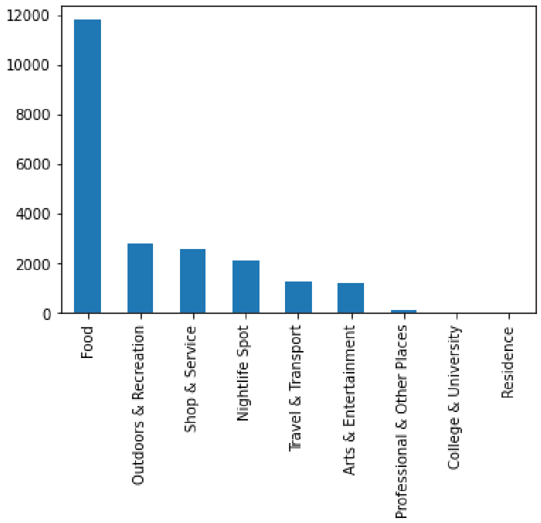


Figure 13. Number of venues by primary type in the data obtained

We can see that most of the venues presented are food related. This can explain the big influence that food related venues appear to have over the result, being the densest venues at either the reference postal codes or the most recommended ones.

On the other hand, there are almost no professional, college or residence venues. This can explain the uncorrelation of the results regarding distance from the centre or most dense venue type when using for the recommendation just venue types of those kinds.

4.1.1. Results for density criterium

For each combination of postal code area available and city available to move in, the recommendation algorithm will be used for getting a series of recommendations based on density, one for each of the venue recommendations lists defined to be considered. For each of these recommendations, the following data will be stored in a dataframe:

- User hypothetical input (city to move in and venue categories to consider).
- Distance and density of each of the postal codes considered given by the recommendation.

For checking the hypothesis of the correlation between the distances from the city centre of every postal code and the density of venues in these postal codes, the following info will be show for each of the venue category lists used previously:

- A scatter plot showing the relationship between distance from centre and density for all tests done with these venue category lists to be considered.
- The Pearson coefficient and P-value of the relationship between distance from centre and density for each of these venue category lists to be considered.

The results can be seen in the following page.

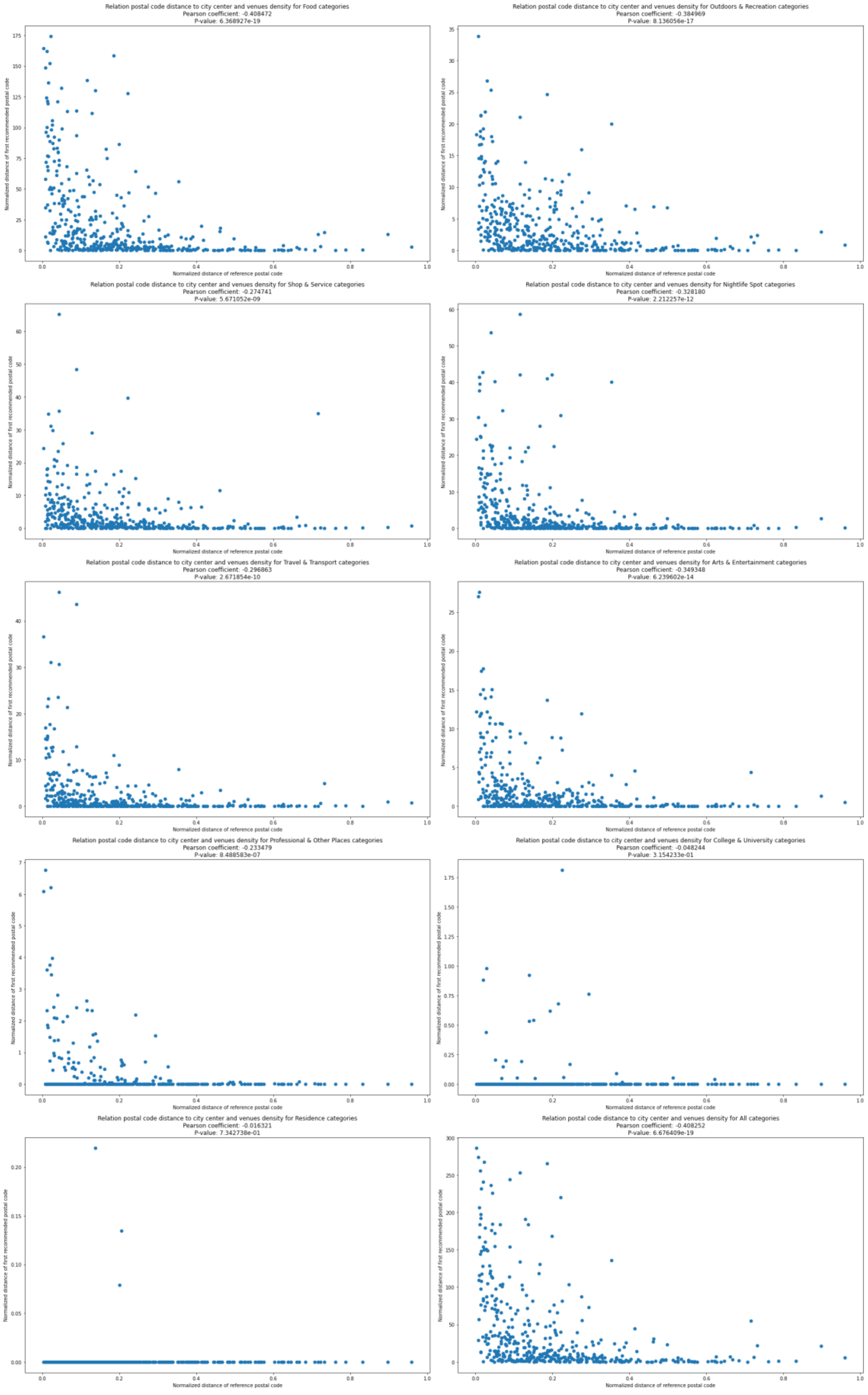


Figure 14. Correlation between distances to centre for similarity criterium

Both the scatter plots and the Pearson correlation values clearly show the following:

- There is a negative moderate correlation between distances when using all categories and food, outdoors, nightlife and arts & entertainment related categories.
- There is a negative weak correlation between distances when using professional, shopping and transport related categories.
- There is no existing correlation between distances when using college and residence related categories.

As with similarity criterium, the P-value at all cases tend to be extremely close to 0 in all cases with some correlation, denying the null hypothesis.

So, it can be considered that there is in general terms a moderate-weak but significant negative correlation between the distances from the city centre of the reference postal code and the density of venues if using food, outdoors, nightlife, arts & entertainment, professional, shopping and transport, and also using all categories.

The histogram of venue types presents in the venues filtered dataset previously shown previously at Figure 14 explain the influence of food categories over the result using all categories (most of venues are food related). and the lack of correlation between distance from centre and density of venues when using college and residence categories (there are almost no venues of those categories).

5. Discussion

After the statistical analysis done in the results section, the following statements have been confirmed when using similarity criterium:

- There is a general positive moderate to weak correlation between the distance from city centre of the reference postal code and the most recommended one. This correlation is inexistent only in the situations where professional, college and residence related venues categories are used for the recommendation.
- There is not a correlation between the densest venue category types in the reference postal conde and the most common venue category types in the recommended postal code. In both cases food related categories are the most common ones.
- Food related categories are the venue categories which has the most influence over the recommendations done, because of the dominance of these categories in the data used for creating the recommendation algorithm.

And the following statements have been confirmed when using density criterium:

- There is a general negative moderate to weak correlation between the distance from city centre of each postal code and the density of venues provided by the recommendation algorithm. This correlation is inexistent only in the situations where college and residence related categories are used for the recommendation.
- Food related categories are the venue categories with the most influence over the density observed in the postal code areas, because of the dominance of these categories in the data used for creating the recommendation algorithm.

Based on these hypotheses confirmed, the content-based recommendation system can be considered as valid for providing recommendations of postal codes where to live based on the preference of the user about venue types he is interested in, cities where the user wants to move in and postal code areas which the user lives now or considers them as fitting well with him.

Nonetheless, it has been observed that the venues related with food (like supermarkets and restaurants) have an excessive influence over the recommendation done, whatever recommendation criterium is chosen. In order to limit the influence of these kind of venues, the following points are proposed as next steps to follow:

- Increment the number of venues obtained in the data obtention section by exploring the use of other databases and APIs. A possibility to be considered is the substitution of the use of Foursquare API by the use of Google Places API, which may be more complete and can provide more and more varied venues.
- It is important to notice that those venues more common in the data obtained, like food, shop and nightlife related venues, tend to be smaller than those venues which are less common in the data obtained, like colleges, residences, parks and professional places. If accurate, or at least approximate, info can be obtained about the size occupied in square meters by each venue (or at least the average size of each venue type), this information can be introduced in the postal codes venues density dataset. This dataset can be based on surface occupied by venues of each type instead of number of venues of each type.

Other next steps to follow in order to improve the recommendation system are the following:

- Include other information not related with venues in order to use it for making the recommendations. This information can be about the demographical statistics of each area, the type of houses in each area or the cost of living in each area.
- Use smaller areas for the recommendations, like for example census sections, which are smaller than postal code areas.
- Provide additional info about each area recommended, like for instance a list of houses available to rent in the most recommended area.
- Add more cities to the recommendation system, either from Spain or abroad.
- Enhance the final application, by implementing a more robust error control and a visualization enhance for making the application more attractive.

6. Conclusions

Up to this point the work done has been the following:

- Data about 15 most populous cities in Spain, postal code areas located in these cities and venues located at these postal code areas has been obtained and processed in order to take the density of each venue category at every postal code area processed.
- The processed data has been used for creating and tuning a content-based recommendation algorithm for getting recommended postal code areas in a target city to move in based on the following:

  - Whether the user prefers to use a recommendation criterium based on similarity with another postal code or based on the density of venues.
  - If similarity criterium chosen, the postal code area to be taken as reference.
  - The target city to move in.
  - The venue categories the user is interested in.
- A visualization tool has been created in the form of functions for using the recommendation algorithm with the target of showing a choropleth map of the target city with the postal codes of that city coloured depending on the recommendation mark given to each of them.
- An application has been created for using the recommendation algorithm and the visualization tools, providing an easy-to-use user interface for getting a recommendation.
- A series of hypotheses have been exposed after an exploratory analysis done with the application, and a series of statistical tools has been used in order to confirm them or denying them.
- The statistical results have been used in order to confirm the hypotheses exposed after the exploratory analysis.
- Based on the exploratory analysis, the hypotheses confirmed and improvement ideas, a series of steps has been proposed to be followed as a work for improving the recommendation system.

The results got confirm the validity of the recommendation system for doing appropriate recommendations to any user interested in moving into another city, basing the recommendation on the venue types presented in the areas recommended.