Spanish postal code content-based recommendation system

Adrián Rejas

IBM Data Science Certificate capstone project

Index

Introduction

- When people need to move into another city, choosing the right area to live in the new city could be a challenge.
  - Limited knowledge
- Postal code areas: universal city areas demarcation
- Project idea:

> Design a content-based recommendation system for choosing the postal code area to live in a new city

Introduction

Similarity option

- Recommends postal code areas more similar to a reference postal code.
- Inputs:
  - Reference postal code.
  - The city to move in.
  - Amenities of interest.

Density option

- Recommends postal code areas with highest density of venues
- Inputs:
  - The city to move in.
  - Amenities of interest.

Introduction

- First approach:
  - 15 most populous cities of Spain.
- Target users:
  - People living in Spain or thinking about moving into Spain.
  - People thinking about moving into another home in the same city or in another one.
  - People who consider the amenities of the area as an important feature for choosing a home.

Index

Data to use

Spanish postal codes info

- Data gathered for each postal code:
  - 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.
- Source (Thanks to Inigo Flores):
  - https://github.com/inigoflores/ds-codigos-postales
- Cities data: 15 most populous cities of Spain.
  - Cities data source: handmade JSON.

Data to use

Venues at each postal code

- Data gathered for each postal code:
  - Number of venues in the area.
  - Venues of the area, with the following info for each of them:
    - Venue name.
    - Venue location.
    - Venue category
- Source:
  - Foursquare REST API

Index

- Introduction
- Data to use
- **Methodology**
- Results
- Discussion
- Conclussions

Methodology

- **Development environment**
- Data obtention
- Data processing
- Build recommendation algorithm
- Build visualization tools
- Build application

Methodology: development environment

- Programming code: Python.
  - Well suited for data science projects.
  - Main libraries used: Pandas, Scipy, Folium, iPython, Geocoder.
- Source code container: Jupyter notebook.
  - Required by the project.
  - Portable.
  - Flexible.
- Running environment: Anaconda Python distribution
  - Runs locally.
  - Cloud-based environments also usable: IMB Watson, LabsCongitive.ie, Kaggle …

Methodology

- Development environment
- **Data obtention**
- Data processing
- Build recommendation algorithm
- Build visualization tools
- Build application

Methodology: data obtention

- Cities data:
  - Created handmade JSON.
  - Transformed into pandas dataframe.
  - Location obtained by ArcGIS Geocoder.
- Postal codes data:
  - CVS and GeoJSON files got from https://github.com/inigoflores/ds-codigos-postales.
  - CVS transformed into pandas dataframe.
    - Filtered to get only postal codes from previous cities.
  - GeoJSON files stored for future use.
    - Filtered to get only postal codes boundaries from previous cities.

Methodology: data obtention

- Venues data:
  - List of venue categories:
    - Call to Foursquare API categories endpoint.
  - List of venues at each postal code area:
    - Paginated alls to Foursquare API explore endpoint.
    - Search with location and radius of each postal code.
    - Save venues ID, categories, location and related postal code.
  - Search venues for each postal code area.
  - Gather all venues info into pandas dataframe.

Methodology

- Development environment
- Data obtention
- **Data processing**
- Build recommendation algorithm
- Build visualization tools
- Build application

Methodology: data processing

- Data necessary for recommendation system: density of venue types per postal code area.
- Filter venues dataset:
  - Remove repeated venues.
  - Remove venues from not-processed postal code areas.
- Get postal code venues density dataset:
  - One-hot encoding of venues dataset.
  - Sum number of venues of each category for each postal code.
  - Divide by area in square kilometre for each postal code.

Methodology

- Development environment
- Data obtention
- Data processing
- **Build recommendation algorithm**
- Build visualization tools
- Build application

Methodology: recommendation algorithm

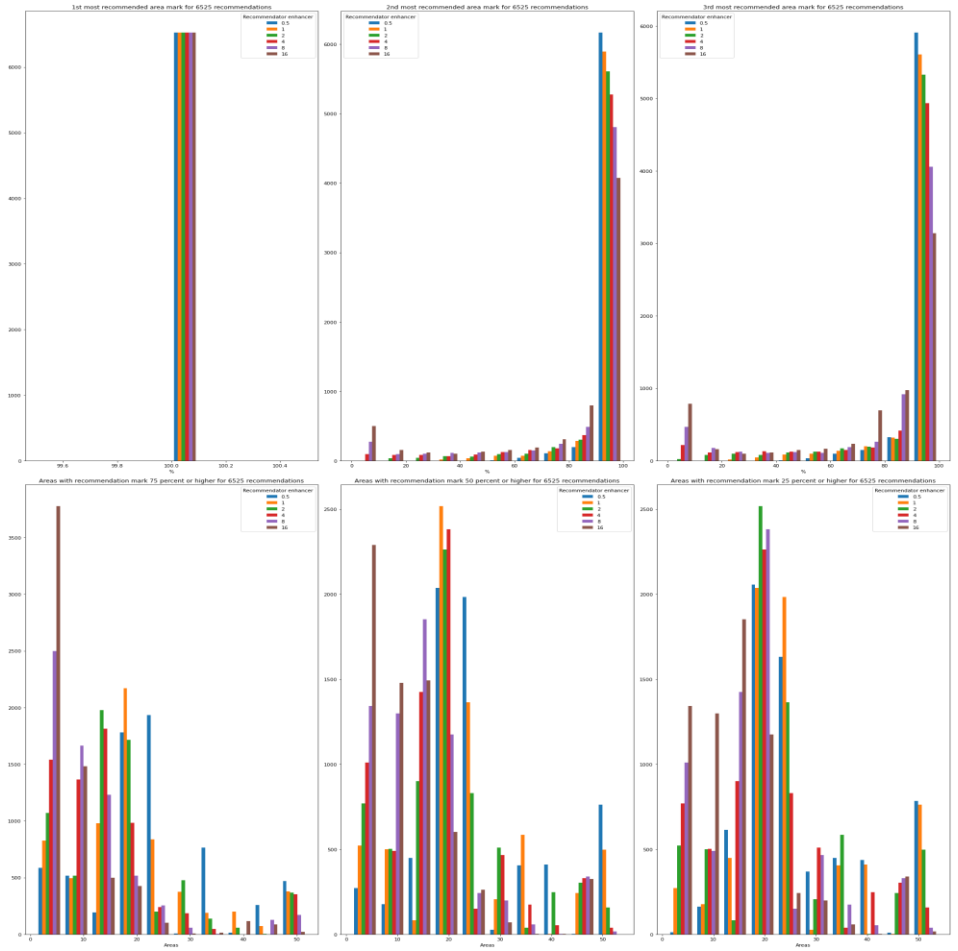
- Algorithm implemented as a function. Inputs:
  - criterium: similarity or density.
  - dataframe: Dataframe with the venues density of the postal codes to be considered.
  - similarity_reference: Maximum distance possible between two postal codes. Used as reference.
  - reference: if similarity criterium used, venues density of the reference postal code
  - venues: venue categories to take into account for the recommendation.
  - Recommendator_enhancer: calibrates the recommendation marks.
  - Similarity_enhancer: if similarity criterium used, calibrates the similarity marks.
- Outputs:
  - Array with recommendation marks.
  - Array with similarity marks if similarity criterium used.
  - Array with densities if density criterium used.

Methodology: recommendation algorithm for similarity

- Custom Nearest Neighbour algorithm.
  - Compares Euclidean distance between postal code areas being considered using density of venue categores.
- Recommendation mark: normalized to maximum value of distances got, inverted and converted to percentage.
  - The highest the better.
  - Calibration through “recommendator_mark”:
    - Between 0 and 1: decrements differences between recommendation marks.
    - Over 1: increments differences between recommendation marks.
- Similarity mark: normalized “similarity_reference”, inverted and converted to percentage.
  - The highest the better.
  - Calibration through “similarity_mark”:
    - Between 0 and 1: decrements differences between similarity marks.
    - Over 1: increments differences between similarity marks.

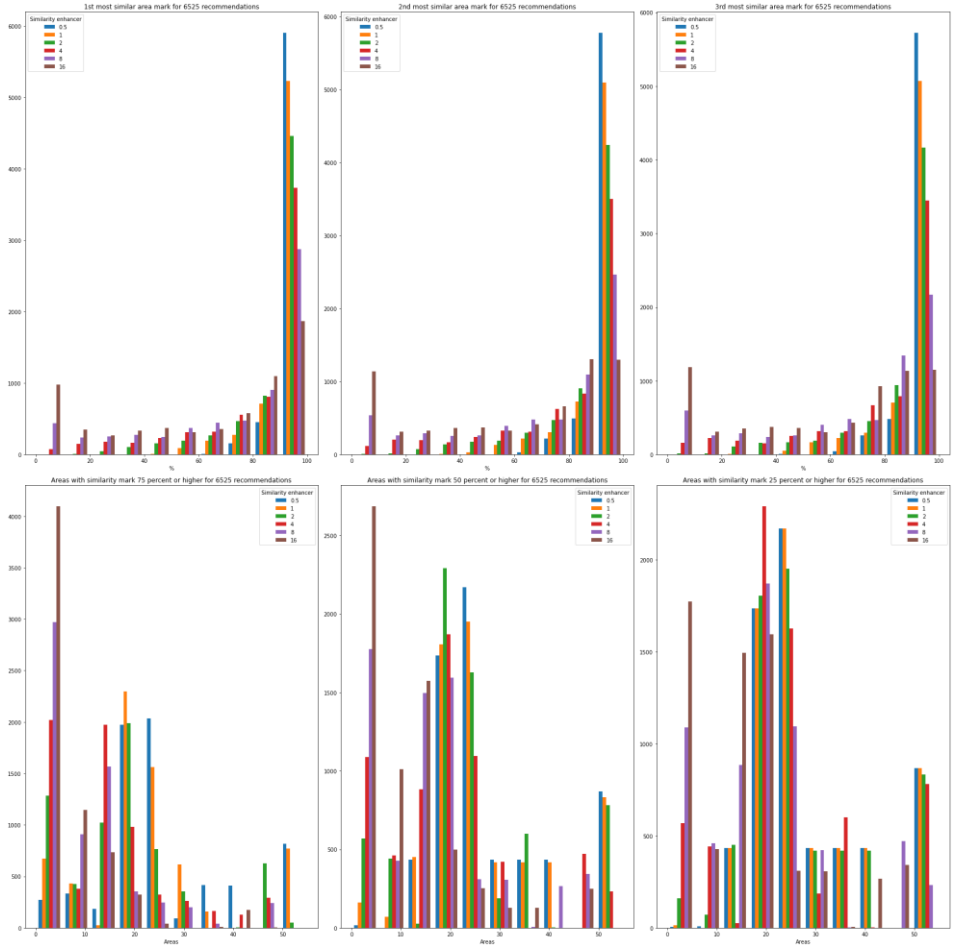
Methodology: recommendation algorithm for similarity

- Testing with different recommendation enhancers.
- Optimum recommendatory enhancer: 8



Methodology: recommendation algorithm for similarity

- Testing with different similarity enhancers.
- Optimum similarity enhancer: 4

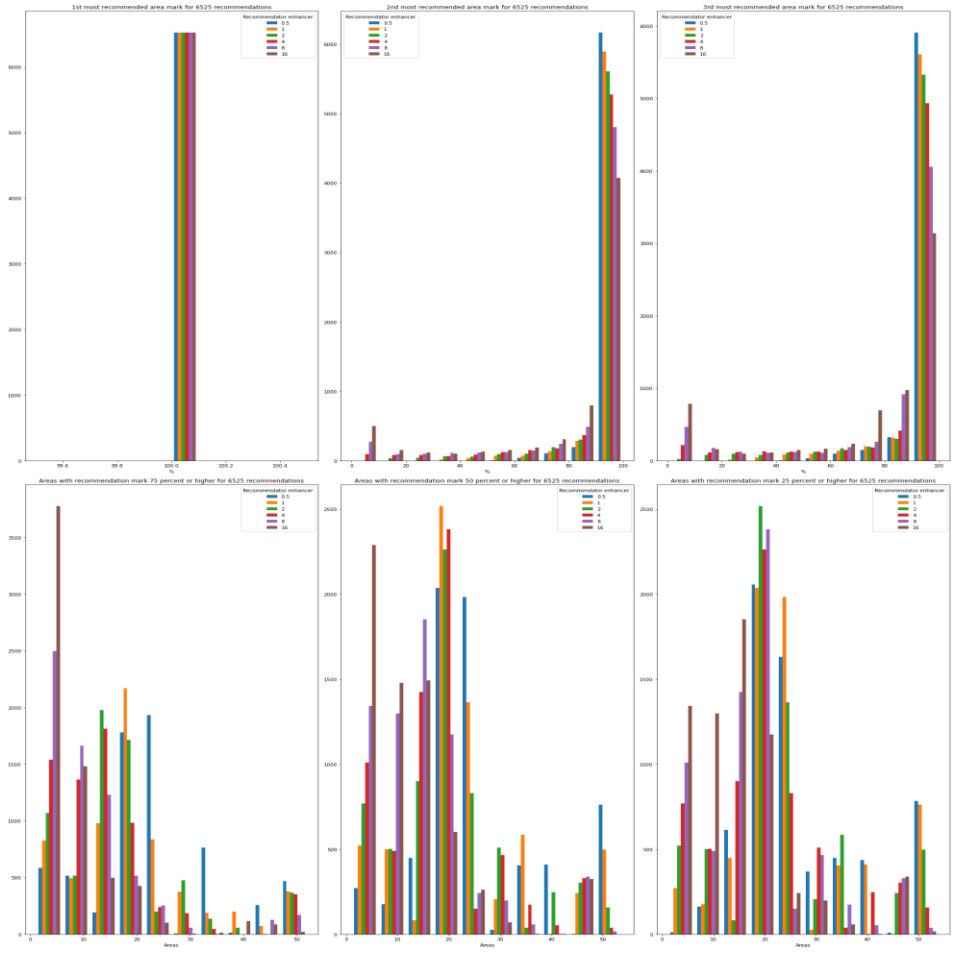


Methodology: recommendation algorithm for density

- Recommendation mark: density of venues normalized to it's maximum, inverted and converted to percentages.
  - The highest the better.
  - Calibration through "recommendator_mark":
    - Between 0 and 1: decrements differences between recommendation marks.
    - Over 1: increments differences between recommendation marks.
- Density: array of density of venues by square kilometre for each considered postal code.

Methodology: recommendation algorithm for density

- Testing with different recommendation enhancers.
- Optimum recommendatory enhancer: 0.5



Methodology

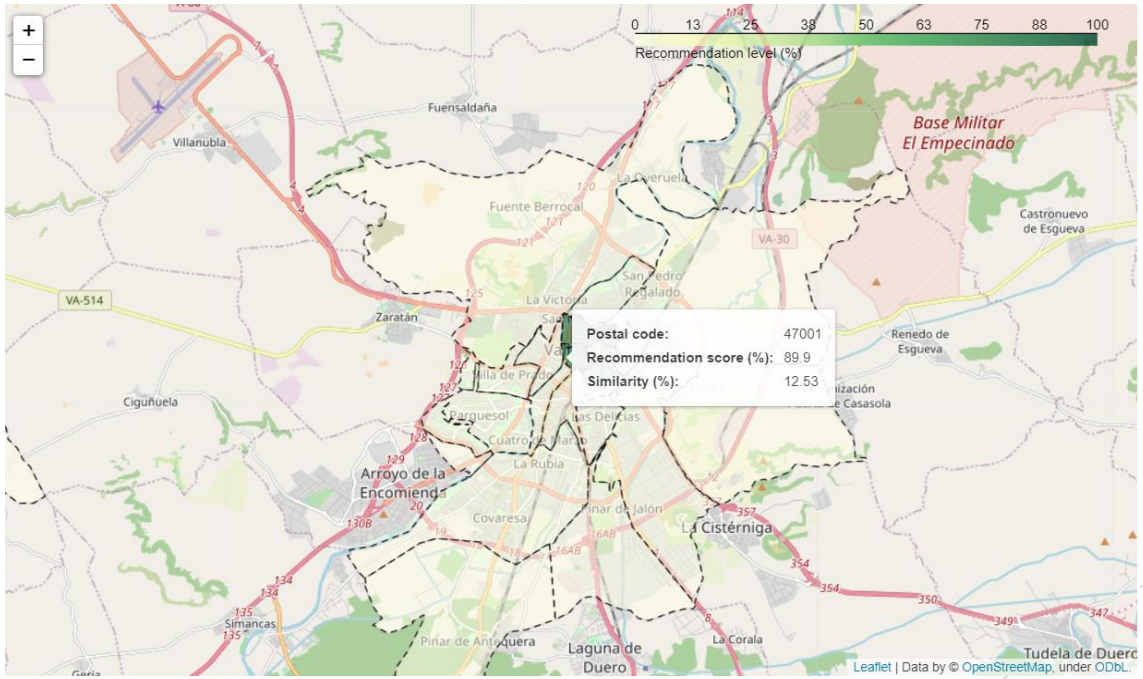
- Development environment
- Data obtention
- Data processing
- Build recommendation algorithm
- **Build visualization tools**
- Build application

Methodology: visualization tools

- Two functions implemented:
  - One for similarity critérium.
  - One for density critérium.
- Both functions will show a choropleth map of a city chosen:
  - Postal codes remarked in green: the darker the more recommended.
  - Info tooltips when mouse placed over postal code area.

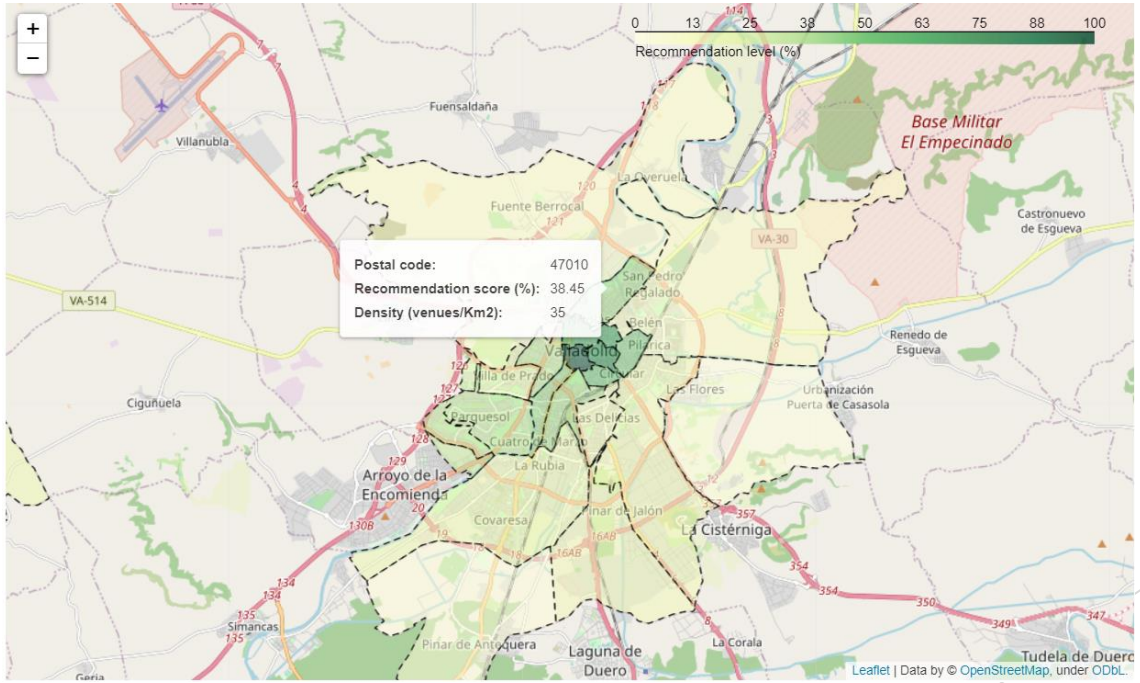
Methodology: visualization tools

- Function for similarity criterium. Input:
  - postal_code.
  - city_to_movein.
  - venues_selected.



Methodology: visualization tools

- Function for density criterium. Input:
  - city_to_movein.
  - venues_selected.



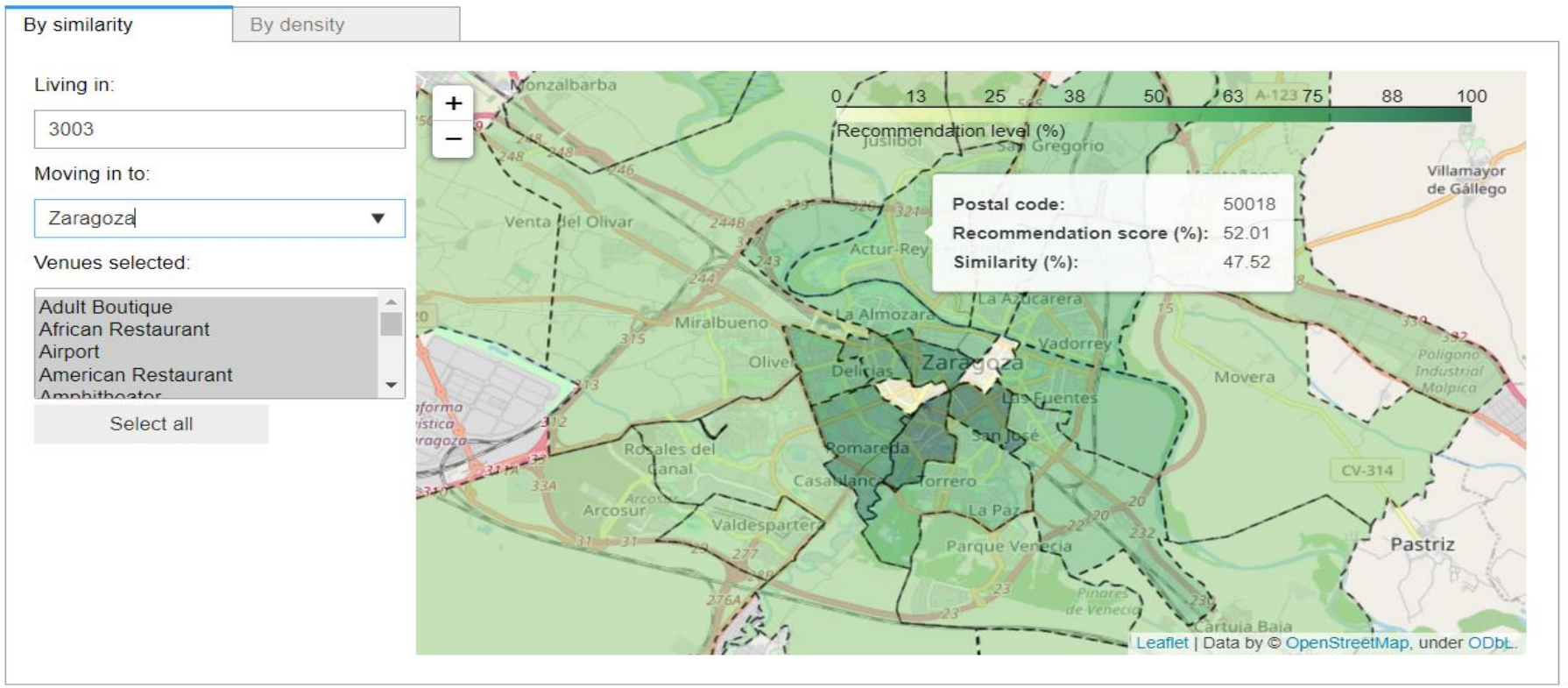
Methodology

- Development environment
- Data obtention
- Data processing
- Build recommendation algorithm
- Build visualization tools
- **Build application**

Methodology: application

- Implemented with Ipython library.
- Use of recommendation algorithm and visualization tools.
- Two tabs, each one for the two criteriums.
- Postal code and city selected with dropdown selectors.
- Venue types selected with multiselectors.
- Basic error handling.

Methodology: application



Index

Results

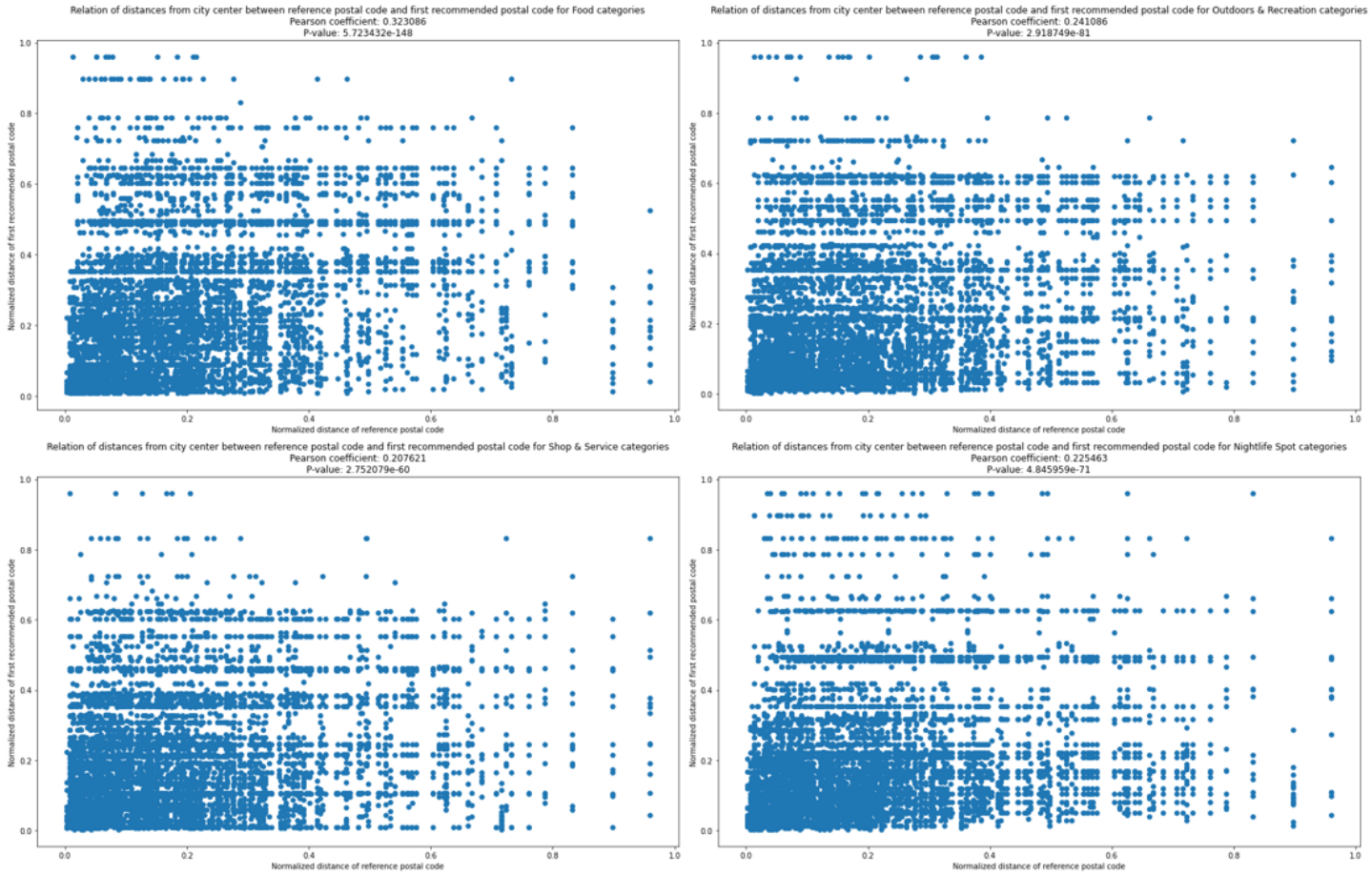
- From exploratory analysis using application, 5 hypothesis exposed:
  - When using the similarity criterium:
    - Correlation between the distances to city centre of the reference postal code and the most recommended one.
    - Correlation between the most common venue type of the reference postal code and the most recommended one.
    - Two previous correlations can vary depending on the type of venues considered.
  - When using the density criterium:
    - Correlation between the density of venues of a postal code and the distance of the postal code from its city centre.
    - Previous correlation, if existing can vary depending on the type of venues considered for doing the recommendation.

Results: Similarity criterium

- Several recommendation dones:
  - One for each combination of reference postal code, target city and 10 posible venue types lists to be considered (one including all).
  - For each recommendation, save:
    - Normalized distance to centre of both reference and most recommended postal codes.
    - Most comon venue types of both reference and most recommended postal codes.

Results: Similarity criterium

- Correlation between distances to data centre (Scatter and correlation):

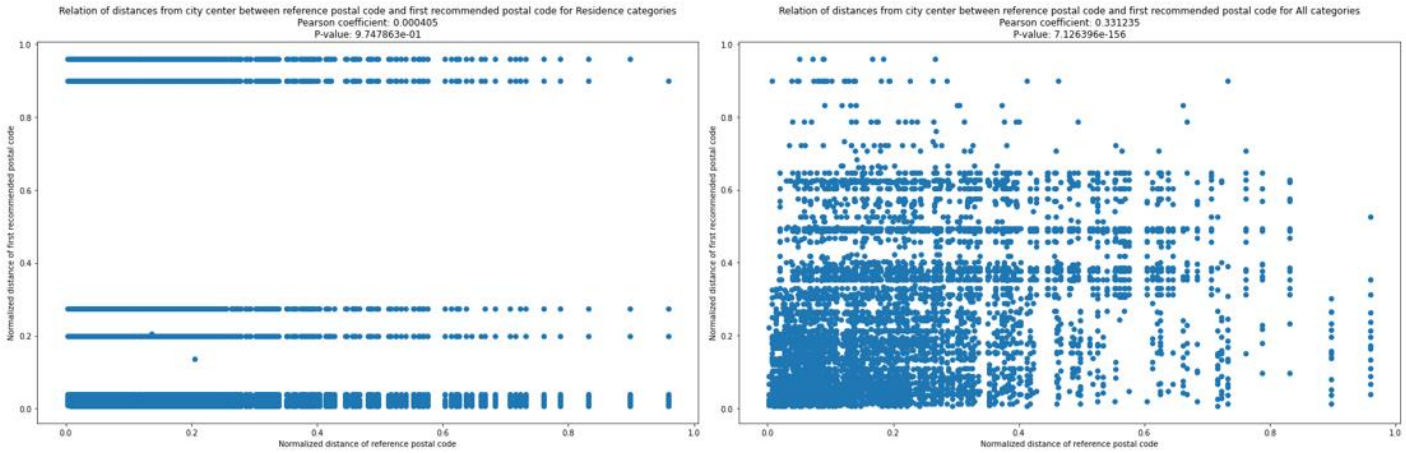


Results: Similarity criterium

- Correlation between distances to data centre (Scatter and correlation) (cont.):

Results: Similarity criterium

- Correlation between distances to data centre (Scatter and correlation) (cont.):



- Interpretation:
  - Moderate to weak but significant positive correlation between the distances from the city centre of the reference postal code and the most recommended postal code when all categories and most of category types lists tested.
  - No correlation using only professional, college and residence categories.

Results: Similarity criterium

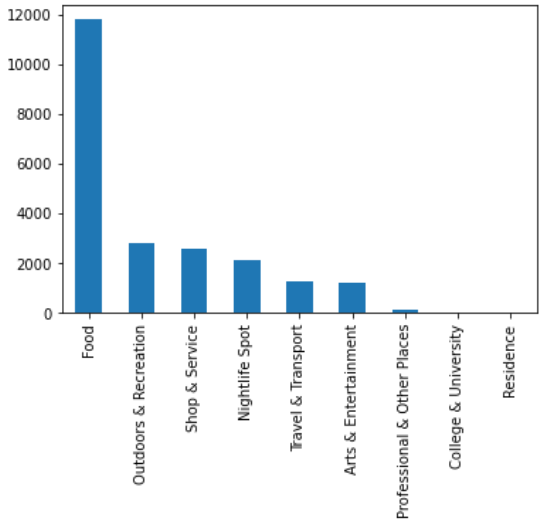
- Check correlation between most common venues (Chi square test):

most_dense_first_catogory_first_recommendation / most_dense_first_catogory_reference	Arts & Entertainment	College & University	Food	Nightlife Spot	Outdoors & Recreation	Shop & Service	Travel & Transport	All
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

- Interpretation:
  - No correlation between most common venues in reference and most recommended postal codes.
  - Food related venues influences more thant others over the result.

Results: Similarity criterium

- Check dependency from venue types (histogram):



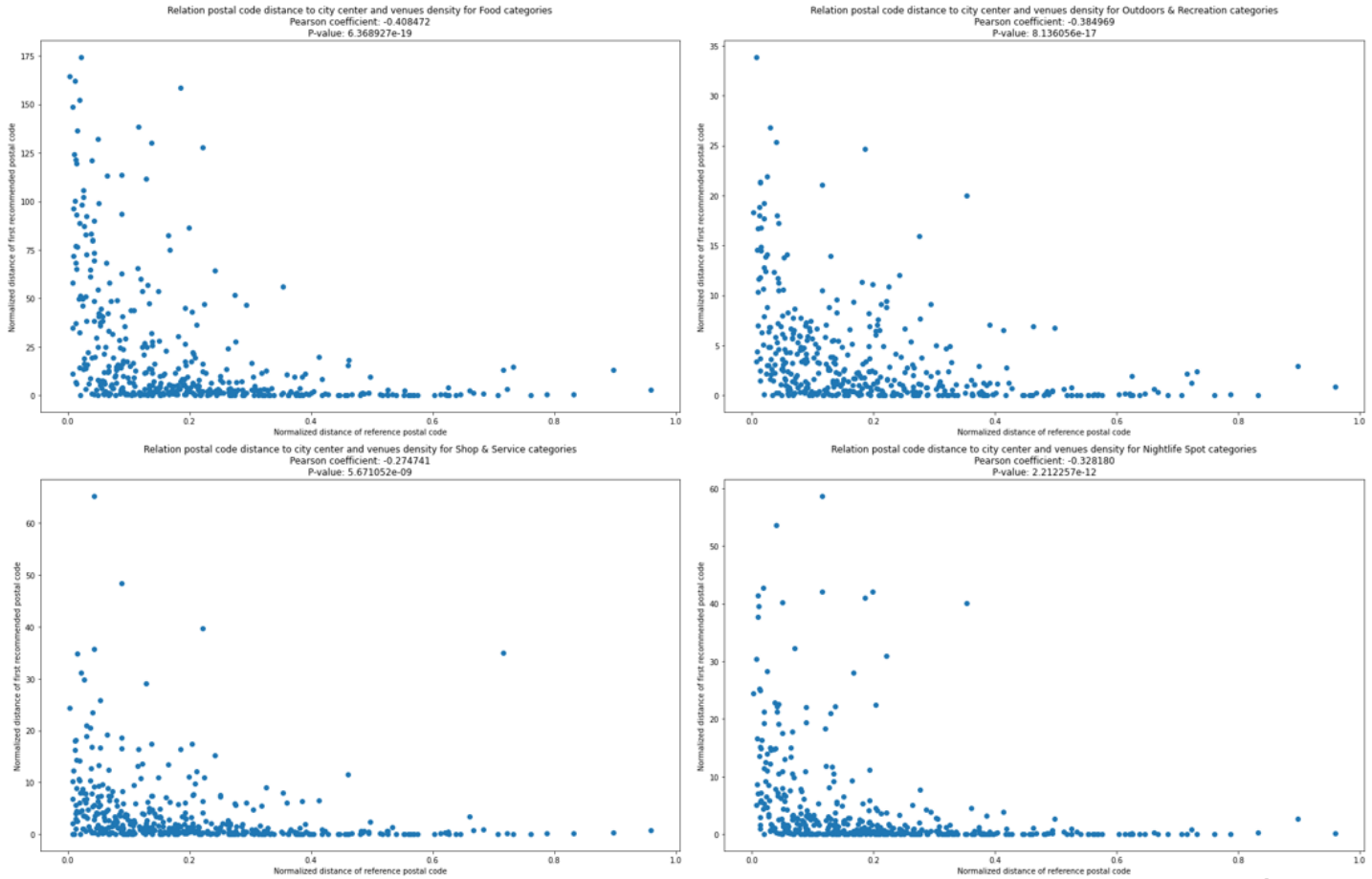
- Interpretation:
  - Food related venues influences more thant others over the result.
  - Professional, college and residence related venues have almost no influence over the result.

Results: Density criterium

- Several recommendation dones:
  - One for each combination of target city and 10 posible venue types lists to be considered (one including all).
  - For each recommendation, save:
    - Density of each of the considered postal codes.
    - Normalized distance to centre of each postal code.

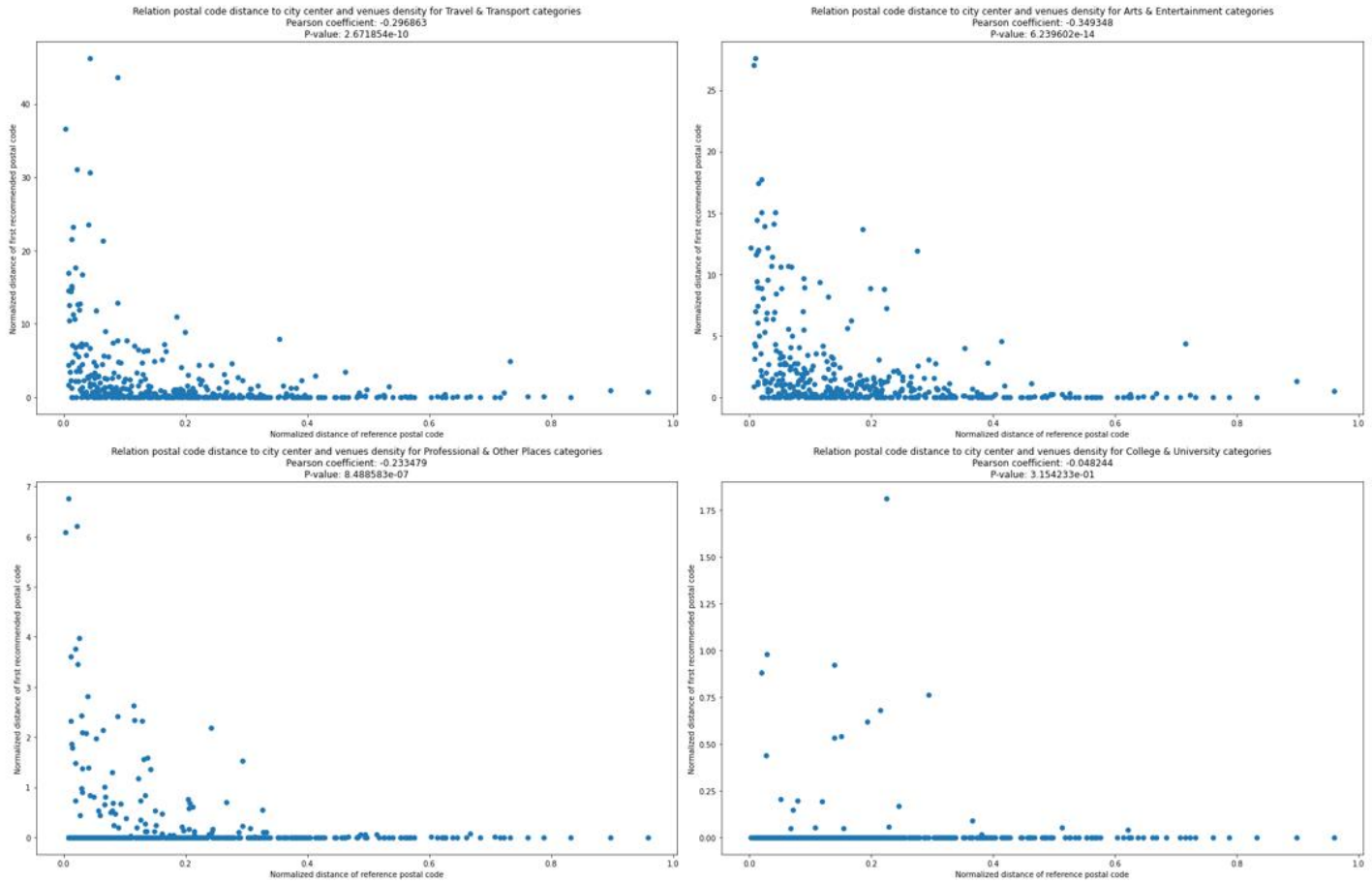
Results: Density criterium

- Correlation between density and distance to data centre (Scatter and correlation):



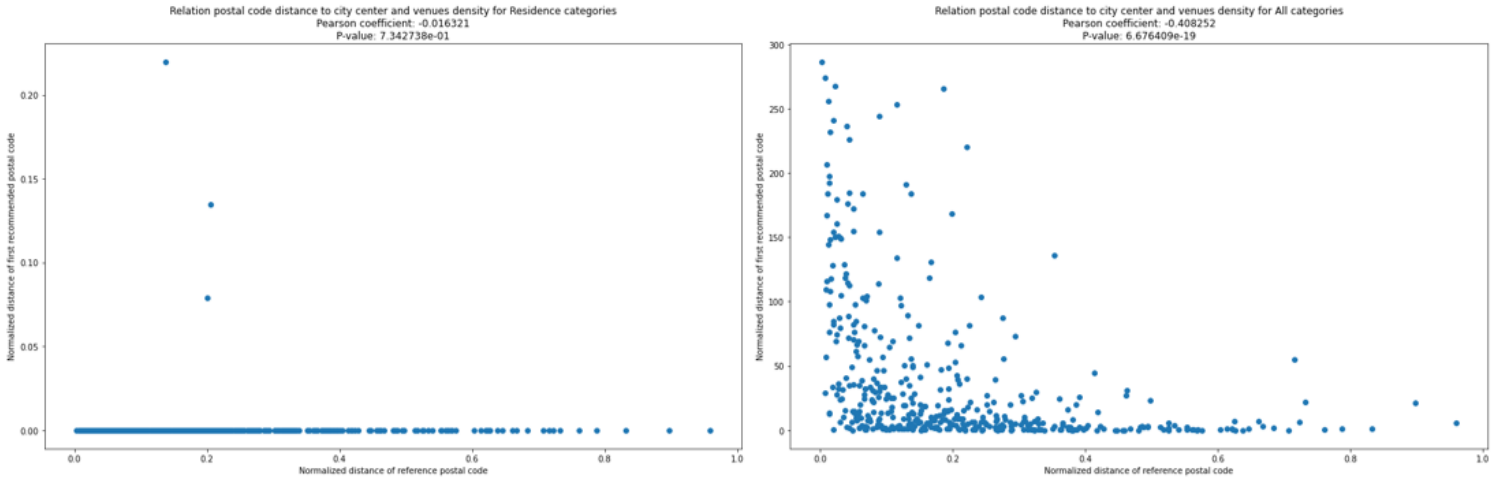
Results: Density criterium

- Correlation between density and distance to data centre (cont.):



Results: Density criterium

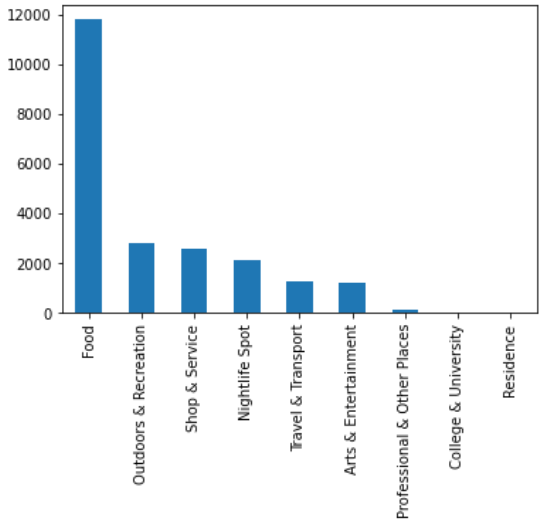
- Correlation between density and distance to data centre (cont.):



- Interpretation:
  - Moderate to weak but significant negative correlation between the distances from the city centre of each postal code and its density in venues/Km2 when all categories and most of category types lists tested.
  - No correlation using only professional, residence categories.

Results: Density criterium

- Check dependency from venue types (histogram):



- Interpretation:
  - Food related venues influences more thant others over the result.
  - Professional, college and residence related venues have almost no influence over the result.

Index

- Introduction
- Data to use
- Methodology
- Results
- **Discussion**
- Conclusions

Discussion

- Statements confirmed for similarity criterium:
  - General moderate-weak positive correlation between distances to data centre of reference postal code and most recommended one.
  - No correlation between most common venue types of reference postal code and most recommended one.
  - Food related venues are the most influential in the recommendation.
- Statements confirmed for density criterium:
  - General moderate-weak negative correlation between density of venues of a postal code and its distance to data centre.
  - Food related venues are the most influential in the density and the recommendation.

Discussion

- General result: Recommendation system is valid.
- Future actions :
  - Increment the number of venues used or use another data source for venues, like Google Places API.
  - Consider the size of the venues for working out the density of venue types.
  - Include other information not related with venues:demographical statistics, the type of houses cost of living ...
  - Use smaller areas for the recommendations.
  - Provide additional info about each area recommended.
  - Add more cities to the recommendation system,:from Spain or abroad.
  - Enhance the visuals and error handling of the application.

Index

Conclusions

- Work done:
  - Data about 15 cities in Spain, postal code areas in these cities and venues at these postal codes obtained and processed for getting density of venues at postal codes.
  - Processed data used for creating content-based recommendation algorithm for getting recommended postal code areas in a target city to move in.
  - Visualization tool has been created: two functions using the recommendation algorithm for showing a choropleth map of the target city with the postal codes recommendations.
  - Application created: provides easy-to-use user interface for getting a recommendation.
  - Hypotheses exposed after an exploratory analysis and checked with statistical tools.
  - Statistical results used for confirming or denying the hypotheses exposed.
  - Previous work used for confirming the validity of the recommendation system and propose future work.
- Final conclusión: Recommendation system created is valid

Thanks for your time