Clustering of houses for sale and for rent in Miami

1. Introduction

Background

In terms of area, Miami is one of the smallest big cities in the United States. According to the country's Census Bureau, the city covers a total area of 143.15 km². Of this area, 92.68 km² are land and 50.73 km² are water. That means Miami is home to more than 400,000 people in 91 km², making it one of the most densely populated cities in the country, along with New York City, San Francisco, and Chicago, among others. The city proper is home to fewer than 1 in 13 South Florida residents. Additionally, 52% of Miami-Dade County's population does not live in any incorporated city. Miami is the only city in the United States bordered by two national parks, Everglades National Park to the west and Vizcaíno National Park to the east.

Problem

With the ever-increasing number of people moving to the city of Miami and the number of housing options available, it is very difficult to determine what types of rentals and sales can satisfy each person. This includes the price of the rental or purchase of housing, number of rooms and bathrooms, places close to the house such as restaurants, parks, shops, gyms, banks, among others. The problem is based on categorizing the homes by the characteristics mentioned above so that anyone can choose among all the options the one that best meets their needs.

Interest

The problem is aimed at people interested in moving to the city of Miami and want to buy or rent a home.

2. Data acquisition and cleaning

Data sources

The characteristics of each home for sale to rent were scraped and taken from the page '*https://www.zillow.com/homes/Miami*,-FL_rb/', using Python code. The venues near the houses were taken from the Foursquare api.

Data cleaning

The data downloaded from the two sources was joined in a table. some data had to be discarded because the information was incomplete or had none and for the purpose of the problem did not provide enough information for categorization.

The data such as the number of bathrooms, rooms, cost and location were contained in a single chain, therefore each of them had to be separated to form a column with each of the values.

Most of the addresses of the houses could not be found through the Geopy api, therefore only the postal code and the city were used to find the places near the house.

Feature selection

After cleaning the data, almost 400 rows of houses for sale and for rent were obtained. the features for each house were the following:

- Latittude and Longitude
- Address
- Status
- Bedroom
- Bathroom
- SQFTS
- Cost
- Restaurants
- Store
- Shop
- Mall
- Gym
- Park
- Bar
- Bank
- Museum

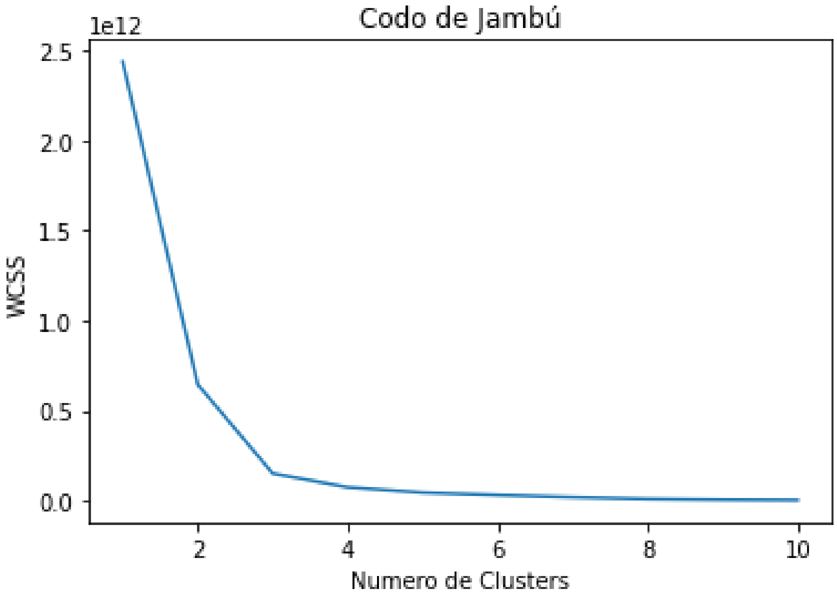
The data was again separated (using the Status column) in houses for rent and houses for sale to make a better categorization for people interested in renting or buying.

3. Explorating Data Analysis

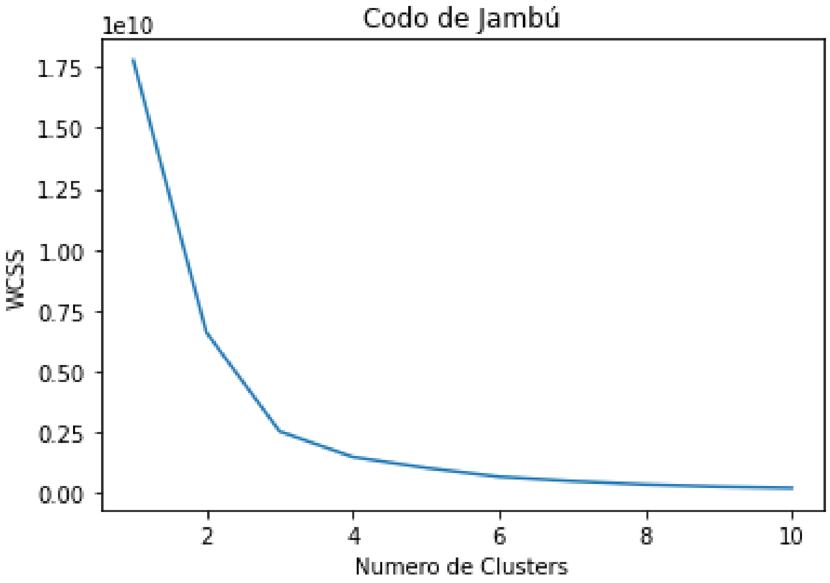
Calculate numbers of clusters

To calculate the number of categories, I used the mean distance of data points to cluster centroid as showed in the next images:

For Sales:



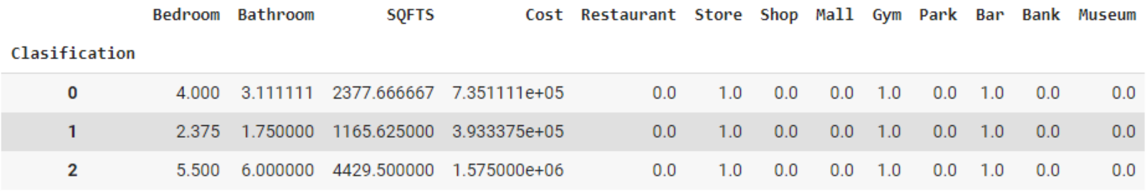
For Rent:



4. Clusters

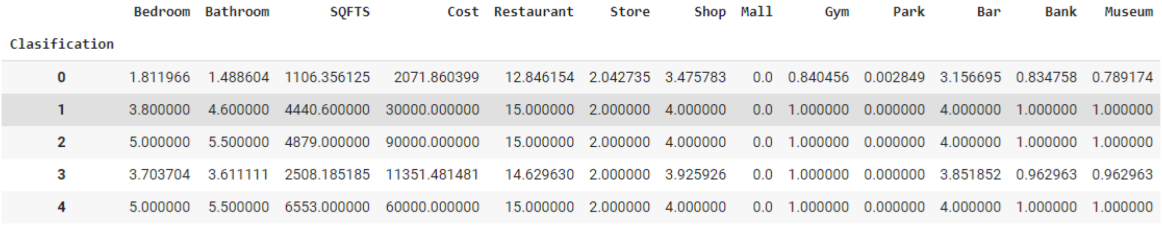
With the model obtained, the following results were obtained:

Three categories for houses for sale. the mean amount for each of its characteristics are as follows



	Bedroom	Bathroom	SQFTS	Cost	Restaurant	Store	Shop	Mall	Gym	Park	Bar	Bank	Museum
Clasification													
0	4.000	3.111111	2377.666667	7.351111e+05	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
1	2.375	1.750000	1165.625000	3.933375e+05	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
2	5.500	6.000000	4429.500000	1.575000e+06	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0

Five categories for houses for rent. the average amount for each of its characteristics are as follows



	Bedroom	Bathroom	SQFTS	Cost	Restaurant	Store	Shop	Mall	Gym	Park	Bar	Bank	Museum
Clasification													
0	1.811966	1.488604	1106.356125	2071.860399	12.846154	2.042735	3.475783	0.0	0.840456	0.002849	3.156695	0.834758	0.789174
1	3.800000	4.600000	4440.600000	30000.000000	15.000000	2.000000	4.000000	0.0	1.000000	0.000000	4.000000	1.000000	1.000000
2	5.000000	5.500000	4879.000000	90000.000000	15.000000	2.000000	4.000000	0.0	1.000000	0.000000	4.000000	1.000000	1.000000
3	3.703704	3.611111	2508.185185	11351.481481	14.629630	2.000000	3.925926	0.0	1.000000	0.000000	3.851852	0.962963	0.962963
4	5.000000	5.500000	6553.000000	60000.000000	15.000000	2.000000	4.000000	0.0	1.000000	0.000000	4.000000	1.000000	1.000000