Sales Territories for Manufacturer in Georgia

Data Science Capstone Project

IN PROGRESS

James Cage
July 2019

Table of Contents

1 Introduction

This is my final project for Coursera's Applied Data Science Capstone class which is the final step in IBM's Data Science Professional Certificate. In this project I refer to a "restaurant equipment manufacturer" and give some details about that company, the market it participates in, and its business strategy. That company is not real. I am not revealing (or for that matter, concealing) any proprietary details about it, because it is a work of fiction and not based in any way on any real company.

I chose this (fictional) project to demonstrate my skills in data gathering, data cleaning (so much data cleaning), machine learning, Python programming, and communicating the results through a report (this), a blog post, and a Jupyter notebook. While the problem described here is a work of fiction, the data, tools, methods, and constraints are entirely real. Nothing has been fudged or ignored in order to make the results more satisfying.

While this scenario is fictional, I am not. My background is in the industrial process control industry, where I worked as an engineer, product manager, and sales consultant. I have some background in inside and outside sales, but I do not pretend to have expert domain knowledge of the restaurant industry. I chose this project in large part because the class required the use of Foursquare data, which is dominated by information about restaurants.

1.1 Acknowledgements

Data in this analysis appears courtesy of Foursquare and its admirable commitment to providing free access to students and individual users. I would also like to thank RestaurantSupply.com of Tempe, Arizona and their inside sales team, which gave me a very valuable overview of the sales process in the restaurant industry. Their help made this a more enjoyable and educational experience for me, but of course any mistakes in the scenario or the project's conclusions are entirely my own.

2 Business Problem

A (fictional) company that manufactures equipment and supplies for Asian restaurants plans to expand into Georgia in the United States. Different types of Asian restaurants use different types of equipment, but the company sells its products to all types. For example, it sells sushi display cases to Japanese restaurants and tandoor ovens to South Asian restaurants. The company wishes to define territories for its salespeople based on the type of restaurants and location in the state. A large concentration of Korean restaurants will hopefully be in a single territory, but that territory might also include a Vietnamese location that is not close to other Vietnamese restaurants.

Sales territories may overlap geographically but given all factors (including the restaurant's cuisine) all territories must be unique. A salesperson may cover Chinese restaurants in one part of the state, while another salesperson would cover all other ethnicities (Japanese, Vietnamese, Korean, etc.) in the same area (assuming there are enough customers in this area to justify multiple salespeople).

The company plans to hire five "outside sales" people who will visit potential customers (restaurants) and sell them on the benefits of the company's products.

2.1 Audience

This analysis will be useful to the following:

- Business-to-business vendors: This analysis will give an example of how to define a customer base, gather data, and use machine learning to segment the base both geographically and by business needs.
- Companies that need a data-driven method to define sales territories.
- People in the restaurant industry who need information on concentrations of venues by type and location.

Data scientists and students faced with the following issues should also be interested:

- Georgia has over 3,000 Asian restaurants. The state covers over 150,000 square kilometers. With my account type, Foursquare limits requests to 100 returned venues in an area no larger than 10,000 square kilometers. This project will demonstrate how to **efficiently gather thousands of Foursquare venues** over a very wide area, while **preventing missed venues and duplications**.
- The problem requires clustering by geographic location and venue type (non-location) data. This project will demonstrate one way to use different types of factors in machine learning to obtain practical results.
- Like most data sets in the real world, the information gathered for this project contains mistakes. A Mediterranean restaurant may be placed in the Asian category, or a restaurant that should be labeled "Japanese" may be placed in the general Asian category. This project will demonstrate how to **clean data programmatically, using key-word searches** within venue categories.
- Finally, this project will give many examples of displaying data on maps, tables, and charts.

3 Data

I use data from Foursquare to create a database of Asian restaurants. Foursquare does not include South Asian restaurants in its "Asian" category, so those were added to the database using "Indian" and "Pakistani" category codes.

3.1 Obtaining Data

As noted in the Introduction, it was necessary to retrieve thousands of venues over a large area. The point-and-radius method of requesting Foursquare data (demonstrated in the class) was not adequate, as it either introduces gaps between the circles or creates duplicates due to overlaps. I obtained the data using a bounding box method instead. This method is limited by the area covered (approximately 10,000 square kilometers) and the number of results returned (no more than 100). I wrote Python code to divide the geographic area into boxes that are small enough for Foursquare queries. When a

bounding box request returns 100 results, my code divides the box into four smaller boxes and submits a new request for each one. This continues (using recursion) until all requests return 99 or fewer results.

The figure below shows an example of the requests (represented by boxes of various sizes) and the number of results (represented by the color of each box). Note that there are no gaps between the boxes and no overlap. The "drop_duplicates" method in Pandas indicate no duplicate venues are included when this method is used.

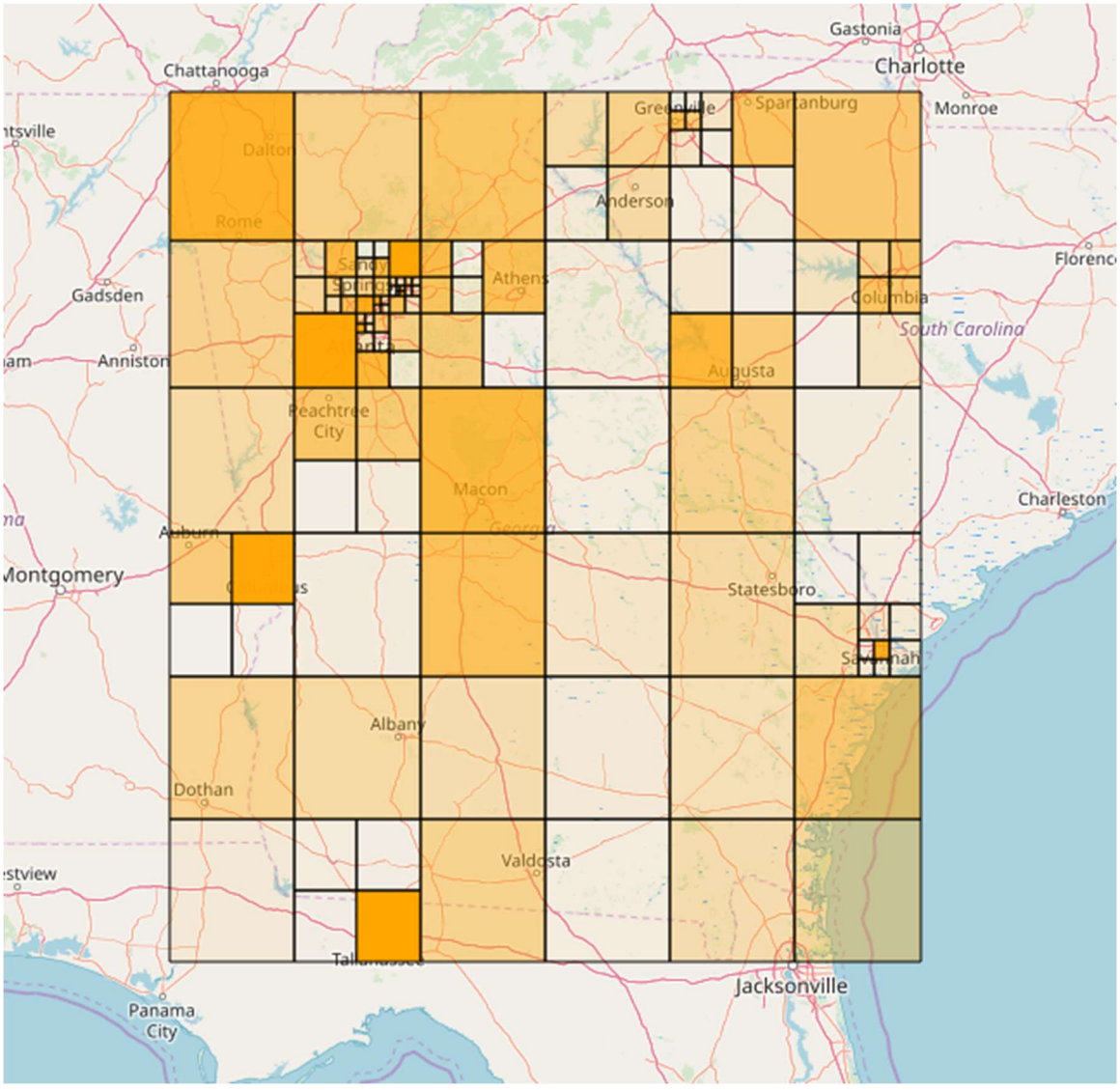

Figure 1: Bounding box Foursquare requests using recursion in Python

The next figure shows some of the smallest boxes in the picture above. These boxes are about 1.4 miles on a side, or roughly 2 square miles. Obtaining this data using equal-sized boxes (or equal-sized circles using the method demonstrated in our class) would require **over 25,000 requests.** My method uses roughly **150 requests** and **executes in less than 90 seconds**. It returns some venues in adjacent states which are dropped during data cleaning.

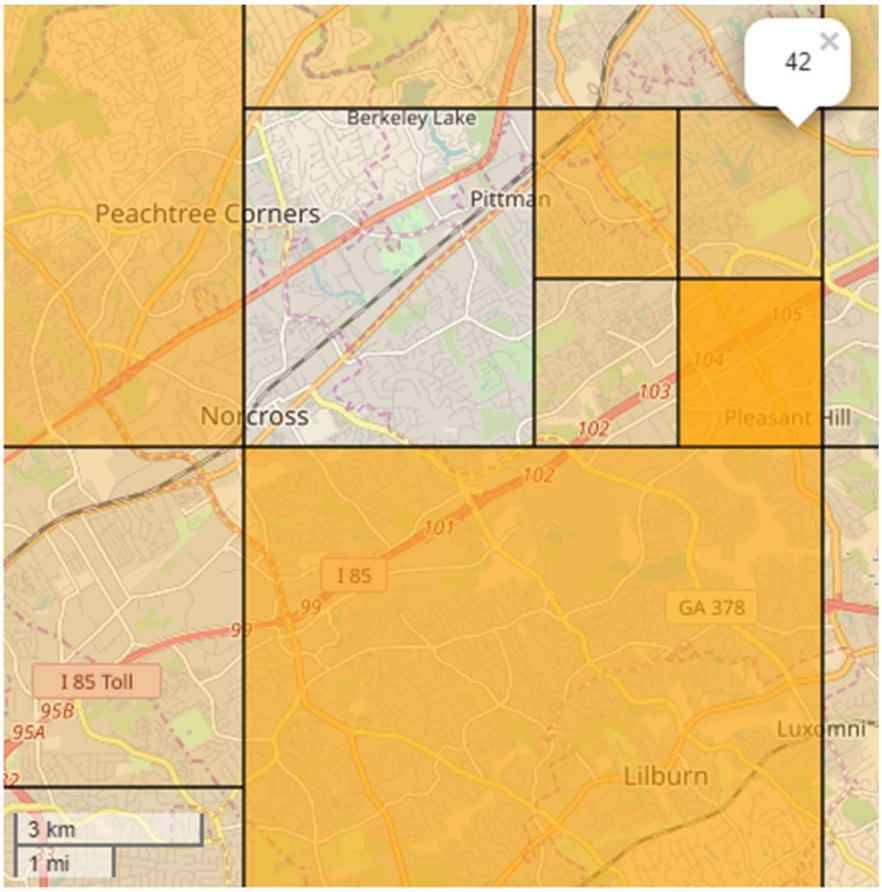

Figure 2: Detail showing size of smallest bounding boxes (scale at lower left)

3.2 Cleaning & Scaling Data

Significant work was required to clean the data. I wrote code to reassign results in incorrect categories and to consolidate categories to a manageable set.

Most Asian restaurants (with the exception of South Asian restaurants) in Foursquare are assigned to sub-categories under “Asian”. “Japanese” and “Korean” are subcategories of “Asian”, for example. However, many restaurants are lumped into the “Asian” category that clearly belong to a sub-category (based on the name of the restaurant). My code reassigns restaurants where possible based on keywords in the restaurant’s name. Other restaurants offer a mix of types and remain in “Asian”.

```
[ ]  # Define terms commonly found in restaurant names for each country of origin.

     china_terms = ['China', 'Chinese', 'Wok', 'Hong Kong', 'Panda', 'Peking',
                    'Beijing', 'Great Wall']
     japan_terms = ['Japanese', 'Tokyo', 'Japan', 'Osaka',
                    'Shogun', 'Fuji', 'Sumo', 'Ichiban', 'Kobe', 'Sakura', 'Ramen',
                    'Teriyaki', 'Ninja', 'Shabu']
     korea_terms = ['Korea', 'Gogi']
     thailand_terms = ['Thai', 'Bangkok']
     vietnam_terms = ['Pho', 'Saigon', 'Viet', 'Banh Mi']
     indopak_terms = ['India', 'Bombay', 'Biryani', 'Naan', 'Masala']

     term_list = [['Chinese Restaurant',china_terms],
                  ['Japanese Restaurant',japan_terms],
                  ['Korean Restaurant', korea_terms],
                  ['Thai Restaurant', thailand_terms],
                  ['Vietnamese Restaurant', vietnam_terms],
                  ['Indo-Pak Restaurant', indopak_terms]]
```

Figure 3: Programmatic approach to data cleaning

Foursquare provides latitude and longitude data, which I scaled to give accurate distance information.

4 Methodology

This problem requires segmenting a market both by geographic data (numeric) and by the restaurant's cuisine (categorical data). Unfortunately, mixing data of different types in a single machine learning is problematic. Categorical data can made numeric (by using one-hot analysis, for example), but how should the machine learning algorithm interpret combining this with X-Y location data? It is easy to imagine the algorithm grouping Indian restaurants that share a common latitude while their longitudes (and hence the physical distance between them) varies widely.

I considered using either geographic data or cuisine data, but grouping on either produces unsatisfying results. Grouping solely on restaurant type produces unwieldy territories. Most sales territories only serve one kind of restaurant, but every territory covers the entire state. Travel time would harm productivity in this case. This analysis also uncovers a charming attribute of k means clustering - its tendency to produce groups of widely different sizes.

Territory:	blue	green	purple	red	yellow	Total
Asian Restaurant			476			476
Chinese Restaurant		1223				1223
Indo-Pak Restaurant					295	295
Japanese Restaurant	800					800
Korean Restaurant				217		217
Thai Restaurant				204		204
Vietnamese Restaurant				122		122
Total	800	1223	476	543	295	3337

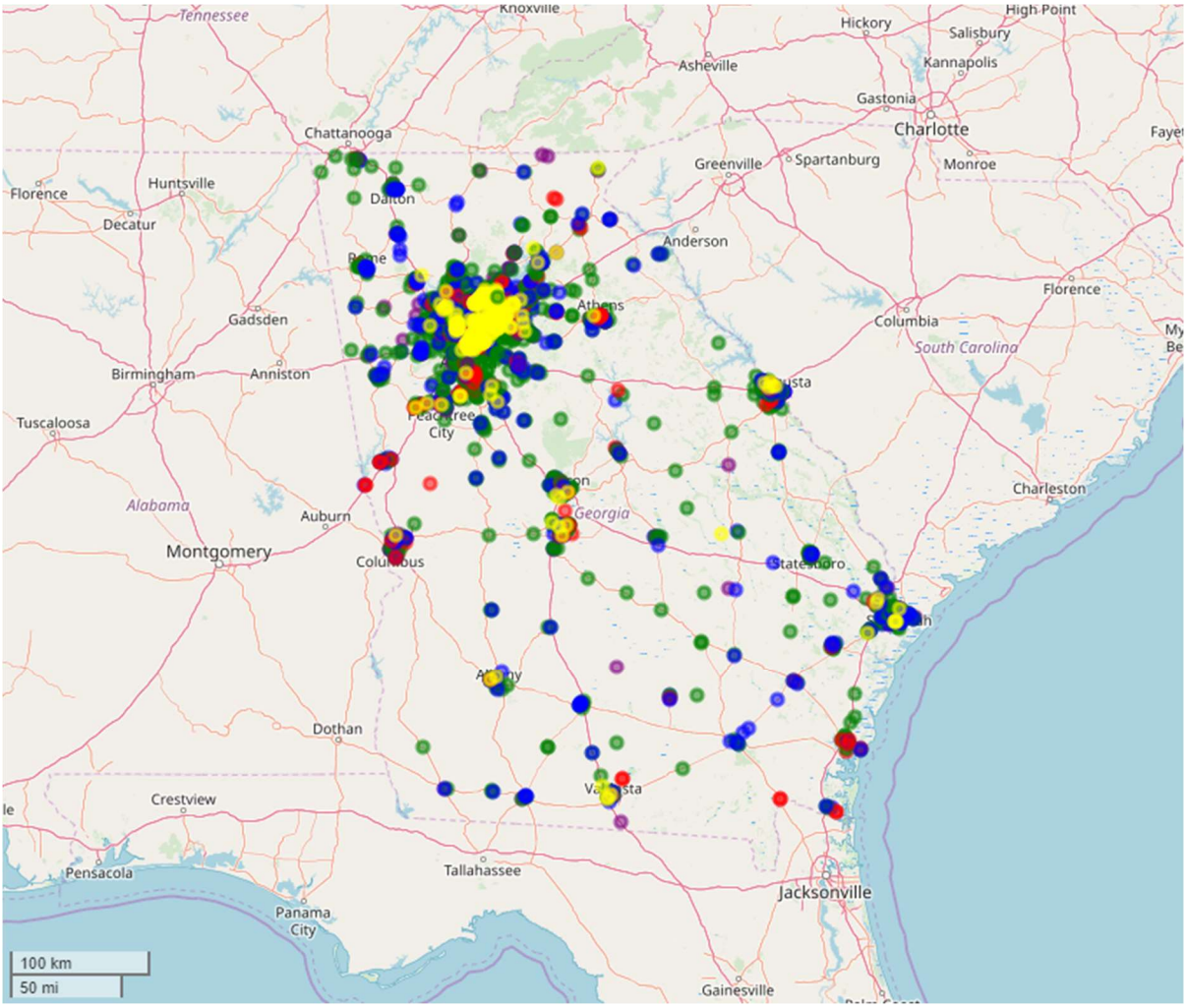

Figure 4: Territories chosen based on cuisine only

Grouping the restaurants physically yields sales territories that include all types of restaurants. This increases the load on the salespeople, as each one must learn to service all types of restaurants. Opportunities to specialize are lost.

Table 1: Description of Table 1

Table	Table	Table

4.1 Sub-Section

Text

4.1.1 Sub-Sub Section

Text

4.1.1.1 Sub^4 Section

Text

4.2 Document Templates Needed:

- Specs
- White Paper
- Reports
- Trade show papers

4.3 Sections needed in this document

1. Revision Information
2. Address / Business Card

4.4 Provide Examples of

- Bullets
- Sub-Bullets
- Tables
- Numbered lists

4.5 Consider for the entire document:

- Use of Color
- Text—should it be indented with the headings?
- Foreground graphic for cover page (world? Hybrid control?)

4.6 Needed in Footer?

- Date
- Author
- Section Title

5 Section

6 Section

6.1 Sub-Section

This is a test of the text in heading 2.

6.2 Sub-Section

6.2.1 Sub-Sub Section

7 Section

8 Section

1. Section 1
    1.1. Section 2
2. Section 3

Appendix 1

Appendix 2