Using Machine Learning for Customer Segmentation: A market study for developing business in the wine industry.

Daniel A. Dycus
ddycus@yahoo.com

This study leverages a dummy client list and feature rich data from FourSquare alongside an unsupervised learning system named DBSCAN to illustrate areas of market potential.

Keywords: FourSquare, geo-location, DBSCAN, machine learning, artificial intelligence

Contents

1 Introduction

This study illustrates the potential client database capabilities of the cloud. Using APIs such as FourSquare we access feature rich data including geolocation coordinates. We leverage this data against a data list acting as a dummy client list. Our clients are wineries and vineyards which are agricultural businesses. All businesses and business locations are stored alongside satellite map data. These businesses rely on suppliers to provide them with commodities. In this study, we demonstrate the ability to access more customers thus increasing market share. We used unsupervised machine learning to identify clusters of business thereby increasing visibility for better market penetration.

2 Backgound

2.1 Backgound

A technical manager has been tasked with generating content in context. In this study, I'll provide a narrative to data we currently have, and data we can use to leverage our position in the global market and increase our market share as a supplier for the wine industry in the North American market.

2.2 Problem

A new technical representative has moved into the Pacific North West. This person is tasked with finding and developing relationships with people in the Pacific North Western United States. How can I help her discover new clients?

2.3 Interest

This brief study represents continued market analysis for an international product development division.

3 Data Acquisition and Cleaning

The sample set we use as the dummy client set was originally found on the web. We imported the dataset using pandas but considered a web scrape and a couple of other data mining techniques. In the end, we found a data-set containing winery data and used Geopy and extrapolated our latitude and longitude coordinates. This information is not confidential and provides a proof of concept for a business development unit.

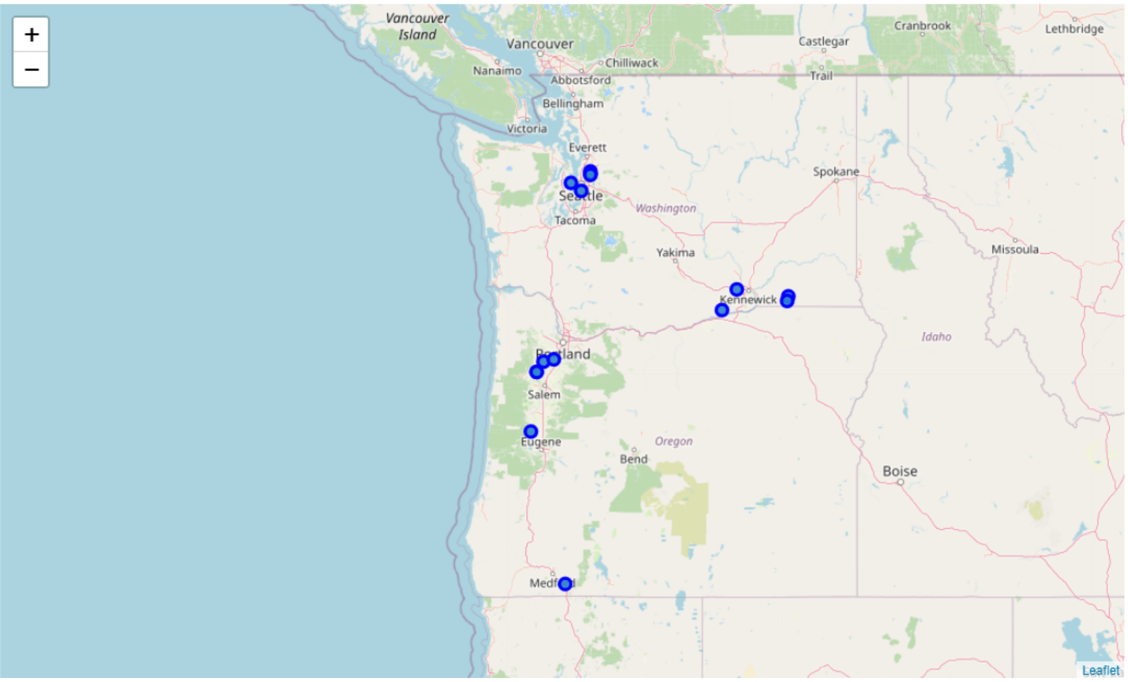

Figure 1: The geo-location data of the dummy client list

3.1 Data Cleaning

We dropped values which were not in the Pacific Northwest from the client list. There were some bad columns and we removed all the erroneous data inside the set we found. Once we had cleaned up our dummy list. We moved to the data coming from the API.

3.2 Feature Selection

The return JSON from the API call is feature rich. We used the built-in category classifier using the SEARCH call. We also included a query for ”Winery”. Our return for the search call only returned categories of vineyard and winery. Other possibilities include venues which are highly rated, popular via check-in data, and those visited by influencers.

We also performed an explore call during our initial investigations to business neighbors. Those results are listed in the Python programming provided alongside this report.

3.3 Data Sources

In conjunction with using the dummy client list, we also leveraged the FourSquare API to explore all the businesses or neighbors of our current clients. Once we had a list of current clients and a list of potential clients we were ready to explore. This represented a high value target acquisition as we illustrate the ability to find all wineries and vineyards respective to any territory on earth.

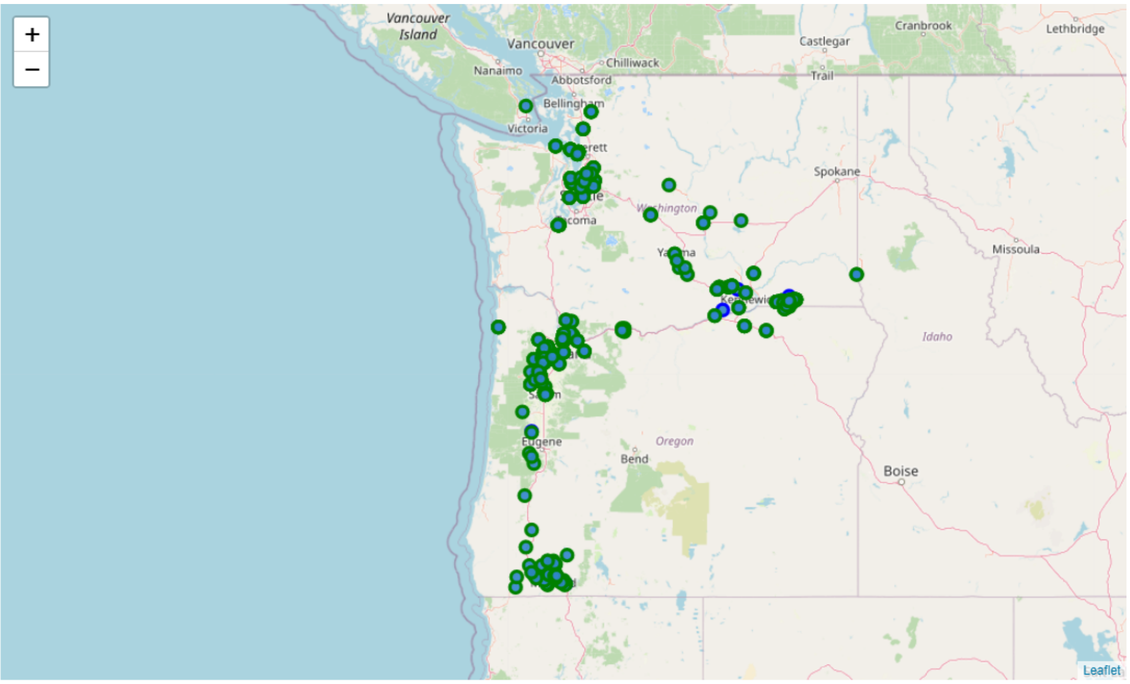

Figure 2: The geo-location data of the potential client list or neighbors of our current clients in the Pacific North West Territory.

4 Machine Learning DBSCAN

DBSCAN: DBSCAN stands for Density-based spatial clustering of applications with noise. DBSCAN is a density based clustering non-parametric algorithm: given a set of points in some space, it groups together points that are closely packed together, marking outliers points that lie alone in low-density regions. DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature. The reasons I chose DBSCAN:

- DBSCAN does not require one to specify the number of clusters in the data a priori, as opposed to k-means.
- DBSCAN can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by (but not connected to) a different cluster. Due to the MinPts parameter, the so-called single-link effect (different clusters being connected by a thin line of points) is reduced
- DBSCAN has a notion of noise, and is robust to outliers.
- DBSCAN requires just two parameters and is mostly insensitive to the ordering of the points in the database. (However, points sitting on the edge of two different clusters might swap cluster membership if the ordering of the points is changed, and the cluster assignment is unique only up to isomorphism).
- DBSCAN is designed for use with databases that can accelerate region queries, e.g. using an R* tree. The parameters minPts and can be set by a domain expert, if the data is well understood.

4.1 How DBSCAN works

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular unsupervised learning method utilized in model building and machine learning algorithms. Before we go any further, we need to define what an unsupervised learning method is. Unsupervised learning methods are when there is no clear objective or outcome we are seeking to find. Instead, we are clustering the data together based on the similarity of observations

4.2 Using DBSCAN for clustering the potential areas for growth

DBSCAN is a clustering method that is used in machine learning to separate clusters of high density from clusters of low density. Given that DBSCAN is a density based clustering algorithm, it does a great job of seeking areas in the data that have a high density of observations, versus areas of the data that are not very dense with observations DBSCAN can sort data into clusters of varying shapes as well, another strong advantage. DBSCAN works as such:

- Divides the dataset into n dimensions.
- For each point in the dataset, DBSCAN forms an n dimensional shape around that data point, and then counts how many data points fall within that shape.
- DBSCAN counts this shape as a cluster. DBSCAN iteratively expands the cluster, by going through each individual point within the cluster, and counting the number of other data points nearby. Take figure 3 as an example.

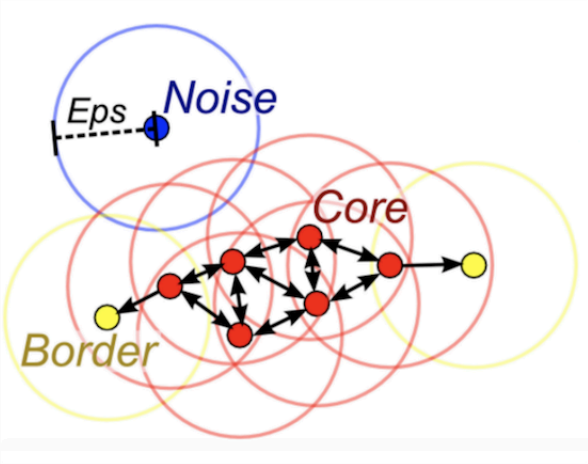

Figure 3: MinPts = 4, Red: Core points, Yellow: Border points and Blue: Noise point

Going through the aforementioned process step-by-step, DBSCAN will start by dividing the data into n dimensions. After DBSCAN has done so, it will start at a random point (in this case lets assume it was one of the red points on Fig.3), and it will count how many other points are nearby. DBSCAN will continue this process until no other data points are nearby, and then it will look to form a second cluster.

As you may have noticed from the graphic, there are a couple parameters and specifications that we need to give DBSCAN before it does its work. The two parameters we need to specify are as such:

- What is the minimum number of data points needed to determine a single cluster?
- How far away can one point be from the next point within the same cluster?

Referring back to Fig.3, *the epsilon (ϵ)* is the radius given to test the distance between data points. If a point falls within the epsilon distance of another point, those two points will be in the same cluster.

Furthermore, *the minimum number of points* needed is set to 4 in this scenario. When going through each data point, as long as DBSCAN finds 4 points within epsilon distance of each other, a cluster is formed.

In order for a point to be considered a *Core point*, it must contain the minimum number of points within epsilon distance. In Fig.3, we only have two core points.

The yellow points on Fig.3 are called *Border points*. These are still part of the cluster because are within ϵ of a core point, but does not meet the minimum number of points criteria.

Finally the blue dots on Fig.3 are *Noise points*. Those are not part of any cluster.

5 Conclusions

This study provides evidence for using machine learning in market analysis and development. Using DBSCAN illustrates areas which are more densely planted as vineyards and possessing wineries. Perhaps most compelling of all is the API which is able to call sophisticated, feature rich data-sets to give market insight into territory developments.

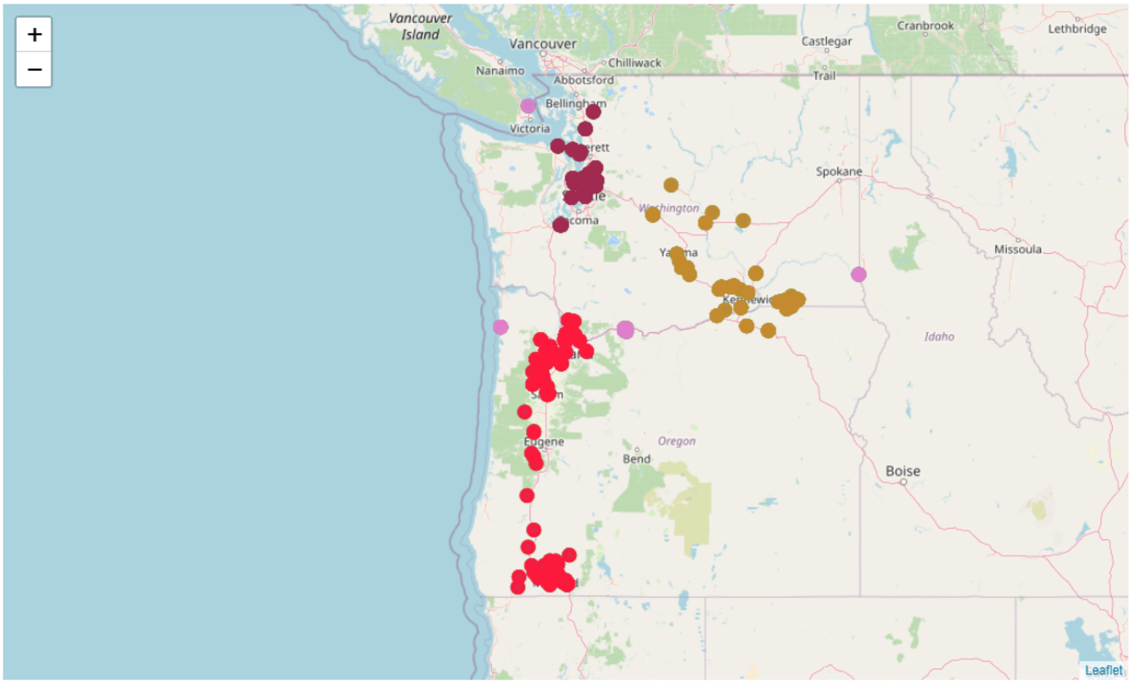

Figure 4: DBSCAN separating clusters from geo-location data

6 Future Directions

The potential client database includes every single winery and vineyard in the world. Developing a strategic global position may be possible using these methods. Future work may include a list of wineries and their contact information for each territory. If content is king, context is god.