Dallas Market Analysis



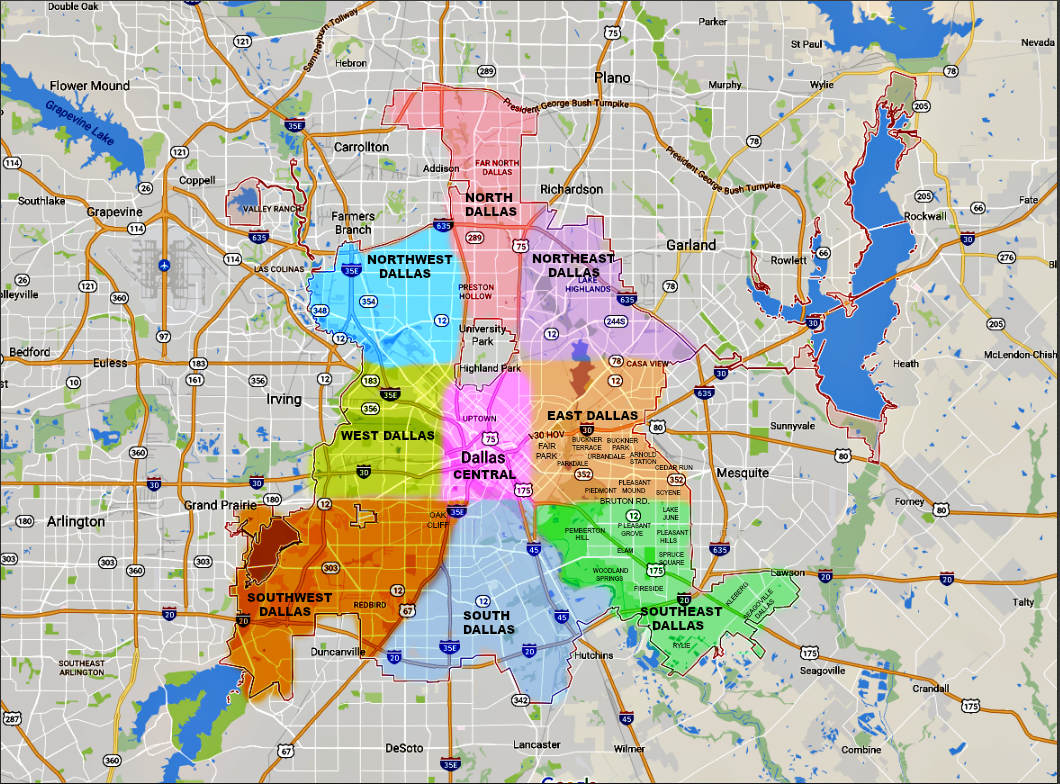
Author: Navie Huynh

Table of Contents

1. Introduction
2. Data Collection and Cleaning
   1. Data Sources
      1. Zip Codes
      2. Four Square API
3. Methodology
4. Results
5. Discussion
6. Conclusion

Introduction

As part of the IBM Data Science Professional Certificate program, we explored New York City and Toronto venues to find hidden insights amongst the different venue locations. The final capstone project requires the learner to utilize data cleaning, preparation, visualization and modeling into their final report. The project will use the Foursquare API to build a data set with the different venues.



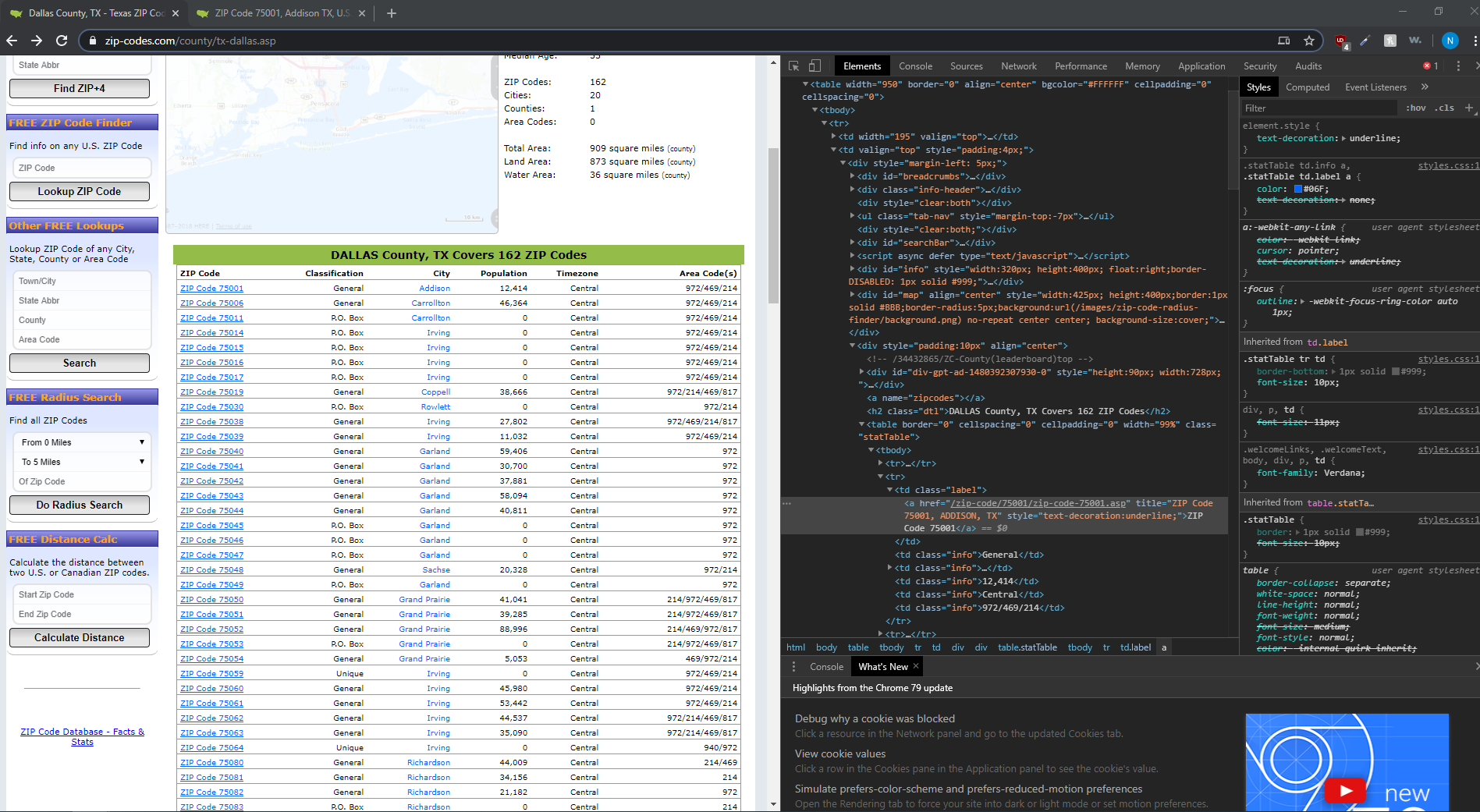
Dallas is the 9th populous city in America. With such a large population, new businesses may not know the prime locations to set up operations. Prime locations include spots where there are few competitors and/or large pedestrian traffic. This paper will seek to identify clusters of markets within the Dallas region.

Key stake holders who may be interested in this paper include companies looking to move operations into Texas. For a company, the environment they operate a business in is key to their success. This paper will explore and compare the different markets within Houston and Dallas, allowing key decision makers to determine the best location for new operations. Other stakeholders may be Investors looking to identify what kind of businesses are hot in specific regions in Dallas. The dataset provided by foursquare will allow us to identify popular venues in each neighborhood.

Data Collection, Cleaning and Exploratory Analysis

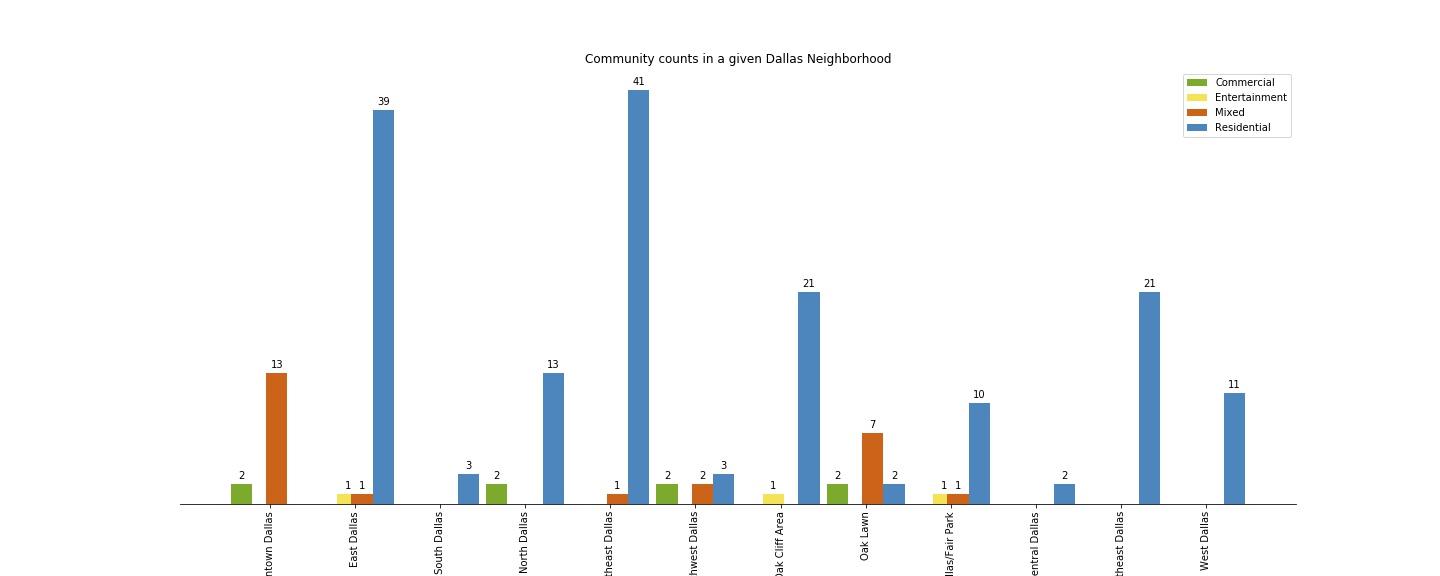
Data Sources:

The first data source that we will use is Wikipedia’s Dallas Neighborhoods page. This will be a good start to exploring the types of neighborhoods in Dallas. During our research, we found that this information, although quality, does not provide us with enough information to explore the markets within Dallas. The biggest challenge in webscraping from Wikipedia was some of the neighborhood tables were embedded in each other, and so figuring out how to create properly formatted tables became an issue.

The main dataset for the project will be from ZipCode.com. The website contains the zipcodes for the Dallas neighborhoods. Some challenges in processing this information was the latitude/longitude data format is not the same across the webpages, and in some cases do not exist. 

Each of the links contains useful geolocation information for a given neighborhood. After collecting this information, we will use the latitude and longitude of each zip code and make requests to the Four Square API, a popular web service that provides useful venue information for locations across the United States. The main challenge in this collection was running out of API calls on a free account and having to wait till the new day to continue the research.

Methodology

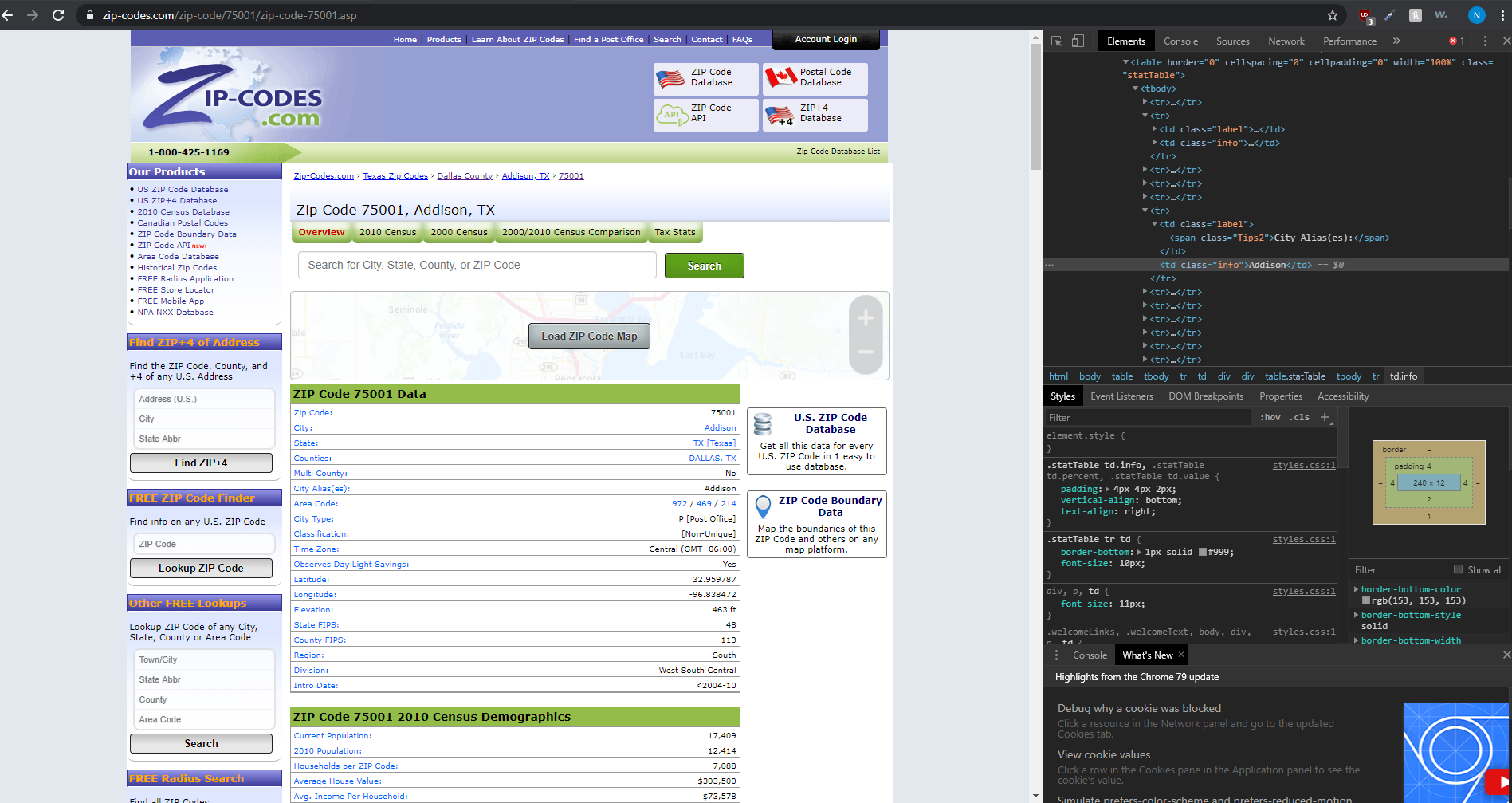
We first began by exploring the Wikipedia page for the Dallas Neighborhoods and seeing what kind of neighborhoods populate Dallas. The main neighborhood types were Residential, Commercial, Mixed and Entertainment. From visualizing the data on Wikipedia, we see most the neighborhoods in Dallas are Residential.

The types of neighborhoods by region allows us to determine optimal locations for specific types of venues, e.g. building a grocery store in Downtown Dallas may not be as optimal as building it in Southeast or West Dallas.

The next website we explore will be zipcodes.com. To scrape the website information, we will use BeautifulSoup, a python webscraping library.



Now for each zip code we will need to scrape the City Alias(es), the Latitude and the Longitude.



We will use the following code snippet to build a webscraper that can navigate to each link and pull the city alias(es), latitude and longitude.

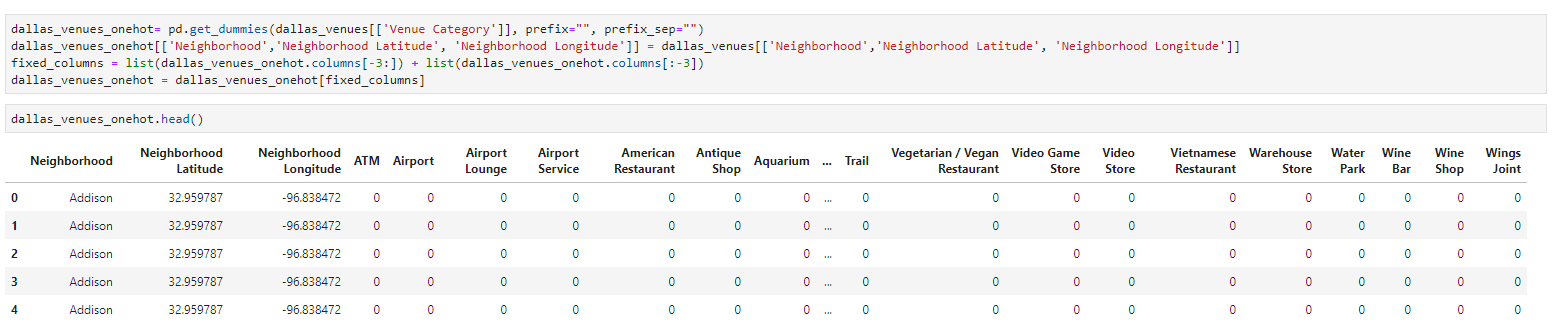


Now that we have the latitude and longitude information, we will make requests to the Foursquare API to find the top 15 venues in each latitude/longitude. Note we will not rely on the Name of the neighborhood as we found that a neighborhood/region can contain multiple zip codes, but a zip code will not contain multiple latitude/longitude.

First, we remove any row with empty latitude/longitude values. We will utilize the requests library to fetch requests of venues for a given latitude/longitude location. Explore my jupyter notebook to see the full code.



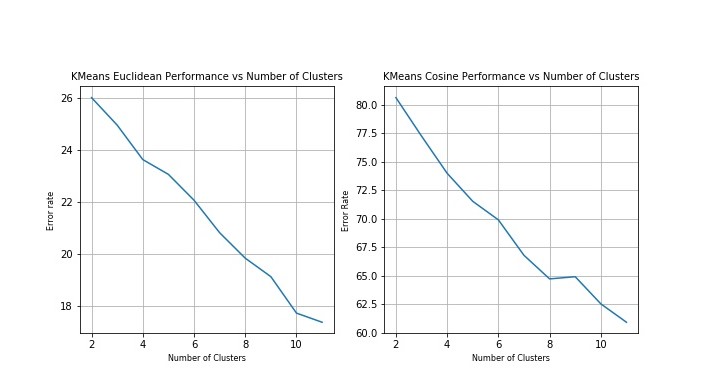
The resulting frame contains about 6000 rows. From here, we apply one-hot encoding to the Venue Category to create a set of features containing the venue type per location.



We will next group the table by Neighborhood Latitude as the Neighborhood Name is not a good feature to merge on per the claim made above. We sort the resulting frame by the frequency of a category and keep the top 15 for our final analysis.

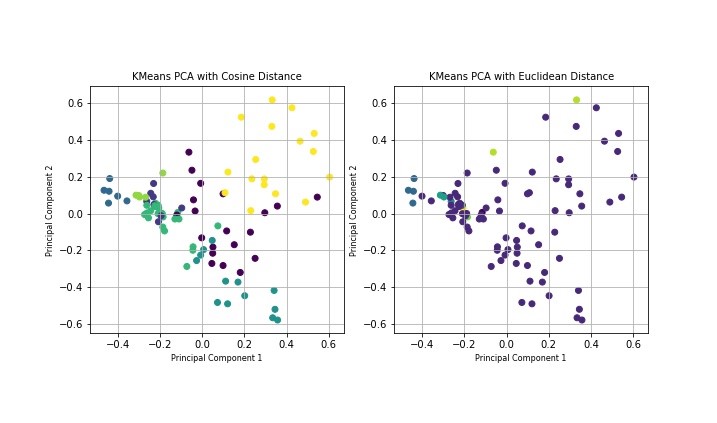


Next we train the KMeans clustering algorithm in two ways to find the best model. The first way we will train the KMeans algorithm is with unnormalized data, also known as using the Euclidean distance as how clusters are measured. What this mean is that the average value is not centered at 0 and the data sample is not normally distributed, i.e. the standard deviation is not 1. The second way we will train the KMeans algorithm is with normalized data. This method is known as using the cosine distances as the metric for how clusters are measured. This method is supposed to optimize the performance of the algorithm for high dimensional problems.



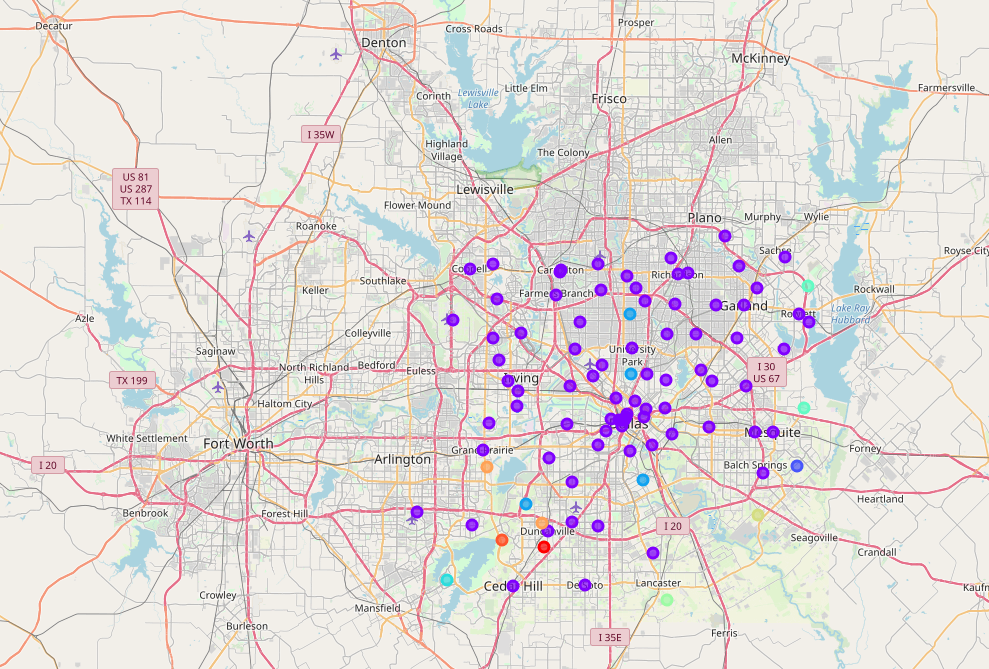
The main task to optimize KMeans is to find the optimal number of clusters. We will train the KMeans algorithm for both Euclidean and Cosine distances for models with clusters ranging from 2 to 12 clusters. We use what is called the elbow method to find the optimal number of clusters. The method seeks to find a bend in the graph of how well the algorithm performs versus the number of clusters and selects the cluster to minimize the diminishing returns, while also avoiding underfitting the data. For Euclidean KMeans, we see that the optimal cluster size appears to be 10, while for the Cosine KMeans we see that the optimal cluster size is about 8.

The next piece of analysis we will perform is principal component analysis. This method seeks to reduce the dimensionality by creating principal components that are linear combinations of the existing features, thus reducing the dimensions needed to analyze while maintaining the important information.



From this visualization of PCA, we can see how the clusters are differently formed when using Cosine distance vs using Euclidean Distance. Though the Cosine Distance appears to distribute the data more uniformly, the performance of the Cosine Distance is worst than Euclidean distance when we measure against the Silhouette score. The metric essentially takes the ratio of the intercluster distance against the intracluster distance as the benchmark for a well performing clustering algorithm. We use the sci-kit learn metric library to calculate the score for both the cosine and Euclidean distance and found that the Euclidean distance scored .2805, whereas the cosine distance scored .1278.

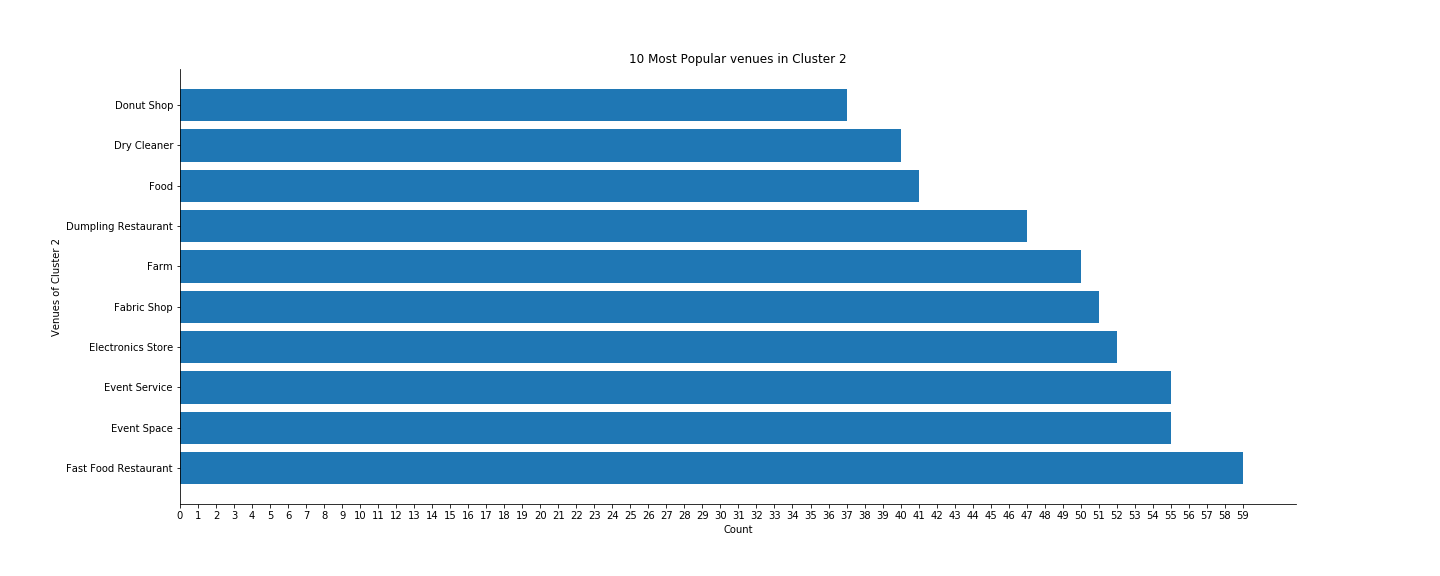
Initially, we ignored the silhouette score and went with the cosine distance as the optimal model, but when analyzing the clusters, we found that they contained very similar venue types, so we selected the KMeans model that uses Euclidean distance with 10 clusters.



Here is a visualization of how our clustering algorithm with Euclidean distance labeled the different neighborhoods (See the notebook for more information).

Results

From our model, many neighborhoods were classified under 1 cluster. One can believe that this a result of insufficient data provided to the training algorithm, but when exploring the cluster in more details, we see that this is not likely to be the cause. What is more likely is that most of these regions in Dallas contain the same types of businesses, which would not be surprising as we saw earlier that Dallas is a mostly residential city.



From analyzing the largest cluster, we see that comfort foods (fast food, donuts and food) are very popular venues in Dallas. Other venues that may be worth exploring are fabric shops and event space/services.

Discussion

From our analysis, we found that most the residence favors comfort foods and entertainment. Starting a business in one of these domains may be hard in terms of existing competitors, but since there is a large market already, the focus can be shifted towards the quality of the goods/services provided.

Another way to break into the Dallas Market may be to figure out how to rebrand the fast food industry into something that is healthier. Texas is a very obese state and the costs of being obese will be a heavy burden on everyone.

The research can be further developed by adding more cities to the database for training. The application of this may help travelers find regions that are as similar as possible to their home location. Another application can be to find other regions where a given franchise may succeed in based on the current location they are headquartered in.

Conclusion

In this paper, we trained a simple KMeans clustering algorithm against web scraped information to determine patterns in the Dallas Neighborhoods to determine what kind of market Dallas has. We found that the most popular venues in Dallas are in the comfort foods category, including Fast food restaurants, donut shops and dumpling shops. The entertainment industry is the next most popular type of venue in Dallas. The algorithm’s performance may not work well in other cities as the Dallas region is heavily biased towards leisure. The performance of the algorithm can be improved with other city venue information.