Segmenting and Clustering Counties for Foreigners in US

Muhammad Haider Raza

May 5, 2020

1. Introduction

1.1. Background

In the context of this analytical study the foreigners refer to people who hold dual nationalities and U.S. Citizenship. Their ancestry originates from a foreign country or foreigners who migrated to and reside in United States. For simplicity we will be focusing on Pakistani Americans. *https://en.wikipedia.org/wiki/Pakistani_Americans [1]*

Typically, when foreigners plan to move to US or even while living in US when they want to relocate to another city/neighborhood/county their preference is to live in a demographic area where nationals from their origin are residing. Apart from that there could be different other factors that influence their decision like distance to workplace, school/college/university for their children, climate, price of property/rent and proximity of amenities.

1.2. Problem

The study here analyzes the available information in this regard and helps take this decision by providing options of similar places (counties) based on an individual preference. There could be so many decisive factors but here I limit my study as per following scenario:

A Pakistani doctor is looking for counties in US where there are considerable numbers of Pakistani households plus there are good number of job opportunities for him that are near to the potential residence which eventually saves him/her commute time. His/her location of interest could be hospitals, clinics, medical research institutes where he can offer his services. We need to present him all the options and cluster similar options together in order to help him take an informed decision.

1.3. Interest

Anyone intending to relocate to (within or from outside) US may naturally have this difficulty in first knowing all the options and then choosing one among all available options as a place of abode. Using Data Science, we can give him an analysis of alike places which could help him choose from the options.

Though there can be several options that could help enhance this search but for my segmenting algorithm following are the main variables to keep the scope simple:

- No. of Pakistani households in a county

- The minimum distance/radius in which he is looking for a job can be an input parameter. For example, here in this analysis we consider a doctor is looking for number of Hospitals available in a radius of 25 Km.
- No. of hospitals, clinics, medical research institutes within a distance (radius) of the county where considerable number of Pakistanis live. Here the place of interest can vary as per his/her profession. Some options could be as given below.
  - Hospital
  - IT Services
  - Research Station
  - Bank
  - University
  - Law School
  - Medical School
  - Trade School
  - School

2. Data acquisition and cleaning

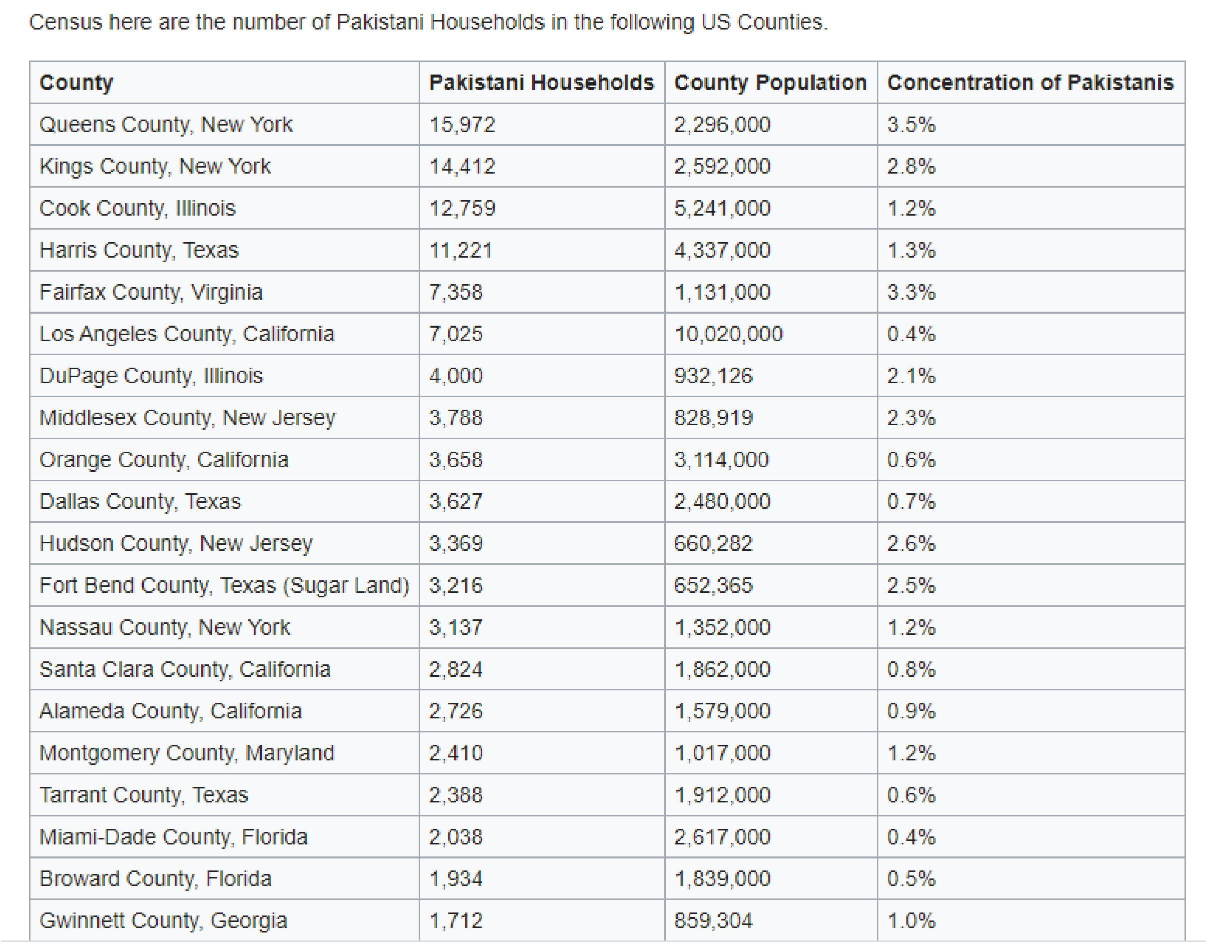
2.1. Data sources

For purpose of this analysis following 3 data sources are used:

2.1.1. Wikipedia web page: Pakistani Americans

https://en.wikipedia.org/wiki/Pakistani_Americans [1]

This has a table which provides a list of counties with number of Pakistani households, total county population and Concentration of Pakistanis.



Census here are the number of Pakistani Households in the following US Counties.

County	Pakistani Households	County Population	Concentration of Pakistanis
Queens County, New York	15,972	2,296,000	3.5%
Kings County, New York	14,412	2,592,000	2.8%
Cook County, Illinois	12,759	5,241,000	1.2%
Harris County, Texas	11,221	4,337,000	1.3%
Fairfax County, Virginia	7,358	1,131,000	3.3%
Los Angeles County, California	7,025	10,020,000	0.4%
DuPage County, Illinois	4,000	932,126	2.1%
Middlesex County, New Jersey	3,788	828,919	2.3%
Orange County, California	3,658	3,114,000	0.6%
Dallas County, Texas	3,627	2,480,000	0.7%
Hudson County, New Jersey	3,369	660,282	2.6%
Fort Bend County, Texas (Sugar Land)	3,216	652,365	2.5%
Nassau County, New York	3,137	1,352,000	1.2%
Santa Clara County, California	2,824	1,862,000	0.8%
Alameda County, California	2,726	1,579,000	0.9%
Montgomery County, Maryland	2,410	1,017,000	1.2%
Tarrant County, Texas	2,388	1,912,000	0.6%
Miami-Dade County, Florida	2,038	2,617,000	0.4%
Broward County, Florida	1,934	1,839,000	0.5%
Gwinnett County, Georgia	1,712	859,304	1.0%

2.1.2. Geocoder

To get the geographical co-ordinates of each county i.e. latitude and longitude, I used
geopy.geocoders python library which converts an address (county name in our case) into latitude and longitude values

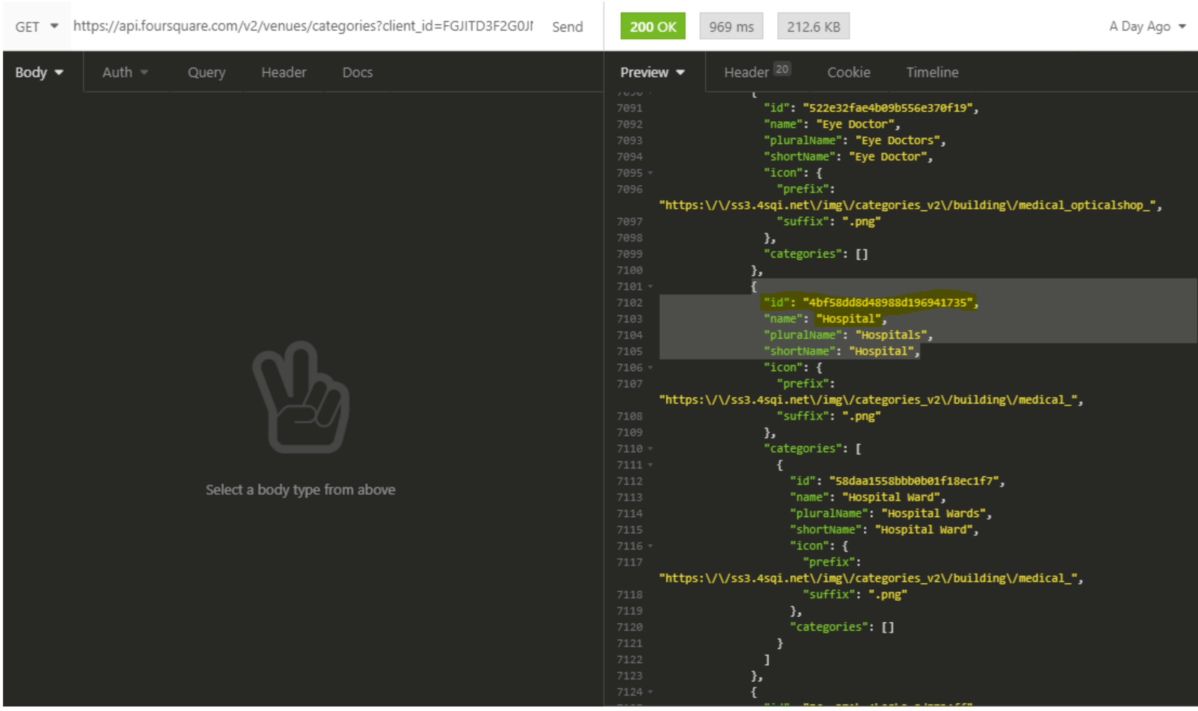
2.1.3. Foursquare

Forsquare APIs were used for two purposes:

- Retrieve list of venue categories

A full list can be obtained by hitting a request to the url:

https://api.foursquare.com/v2/venues/categories?client_id={}&client_secret={}&v=20200430



GET https://api.foursquare.com/v2/venues/categories?client_id=FGJITD3F2G0J

```
                "id": "522e32fae4b09b556e370f19",
                "name": "Eye Doctor",
                "pluralName": "Eye Doctors",
                "shortName": "Eye Doctor",
                "icon": {
                  "prefix":
"https:\/\/ss3.4sqi.net\/img\/categories_v2\/building\/medical_opticalshop_",
                  "suffix": ".png"
                },
                "categories": []
              },
              {
                "id": "4bf58dd8d48988d196941735",
                "name": "Hospital",
                "pluralName": "Hospitals",
                "shortName": "Hospital",
                "icon": {
                  "prefix":
"https:\/\/ss3.4sqi.net\/img\/categories_v2\/building\/medical_",
                  "suffix": ".png"
                },
                "categories": [
                  {
                    "id": "58daa1558bbb0b01f18ec1f7",
                    "name": "Hospital Ward",
                    "pluralName": "Hospital Wards",
                    "shortName": "Hospital Ward",
                    "icon": {
                      "prefix":
"https:\/\/ss3.4sqi.net\/img\/categories_v2\/building\/medical_",
                      "suffix": ".png"
                    },
                    "categories": []
                  }
                ]
              },
              {
```

From above we picked Hospital Category Id to pass in second API call.

- Retrieve venues (Hospital list) around a county

Foursquare APIs are used to find number of venues around a geographical co-ordinate (lat/long of county). Following were important parameters for this API

- CategoryId
- Radius
- Lat/Long
- Intent (equals browse in this case)
- Client Id & Client secret

Used URL: The search endpoint is used to search for venues of a specific type

https://api.foursquare.com/v2/venues/search?intent=browse&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&categoryId={}

2.2. Data Cleaning

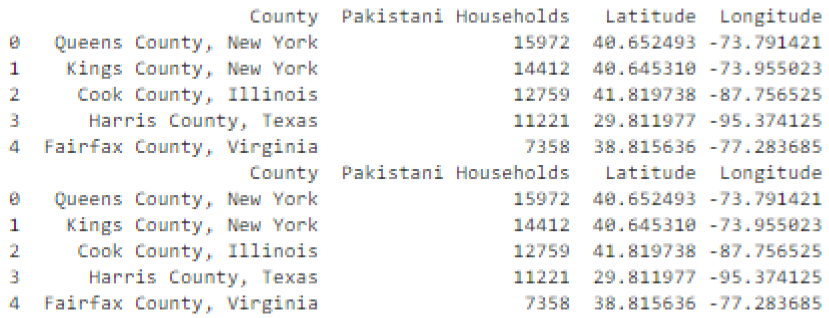
Data downloaded or scarped from multiple sources were combined into a single table.

First of all, the table scraped from Wikipedia page had following columns:

- County
- Pakistani Households
- County Population
- Concentration of Pakistanis

We excluded and dropped following columns from our algorithm to keep things simple: County Population, Concentration of Pakistanis. These could be considered in future enhancements but since we had a column ‘Pakistani Households’ that indicates similar information on strength of Pakistanis living in a county, hence we dropped the column ‘Concentration of Pakistanis’. There were around 33 counties present in the list. Luckily all columns have values present in all the rows.

Next, we needed lat/long information for each county entry in the wiki table so that we can invoke Foursquare API to get the venues count. For this purpose, gecode API was first invoked and the lat/long info was appended to each county resulting in a dataset like:



```
                    County  Pakistani Households   Latitude  Longitude
0     Queens County, New York                15972  40.652493 -73.791421
1      Kings County, New York                14412  40.645310 -73.955023
2       Cook County, Illinois                12759  41.819738 -87.756525
3        Harris County, Texas                11221  29.811977 -95.374125
4   Fairfax County, Virginia                  7358  38.815636 -77.283685
                    County  Pakistani Households   Latitude  Longitude
0     Queens County, New York                15972  40.652493 -73.791421
1      Kings County, New York                14412  40.645310 -73.955023
2       Cook County, Illinois                12759  41.819738 -87.756525
3        Harris County, Texas                11221  29.811977 -95.374125
4   Fairfax County, Virginia                  7358  38.815636 -77.283685
```

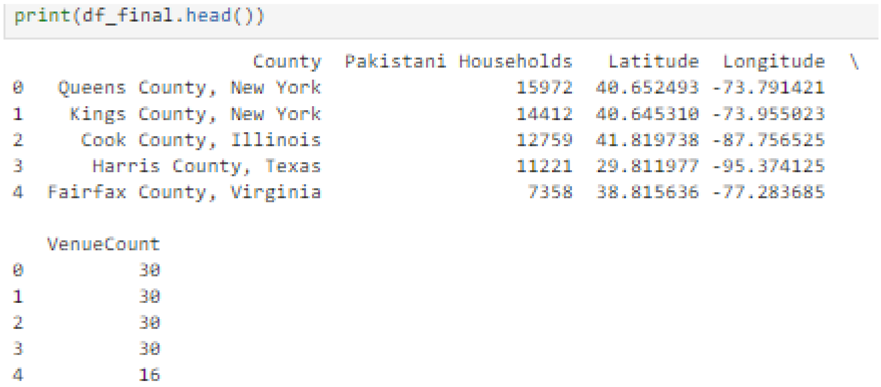
Next, we needed a list of venues for each county within a radius of R (25 Kms) in our case. The venues returned should belong to only the category we are interested e.g. Hospital in current case. For this Foursquare search API was used:

https://api.foursquare.com/v2/venues/search?intent=browse&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&categoryId={}

While invoking foursquare api for each county we observed that for some of the locations the address info was missing. Though this is optional for my scope of study but the issue was fixed by providing a default value of ‘-‘and hence we were able to print the name and address information for each venue without breaking the code.

Since we needed to have the count of venue only we got that by taking the length of the JSON response Array ["response"]['venues']

Now we had a dataframe like:



```
print(df_final.head())
```

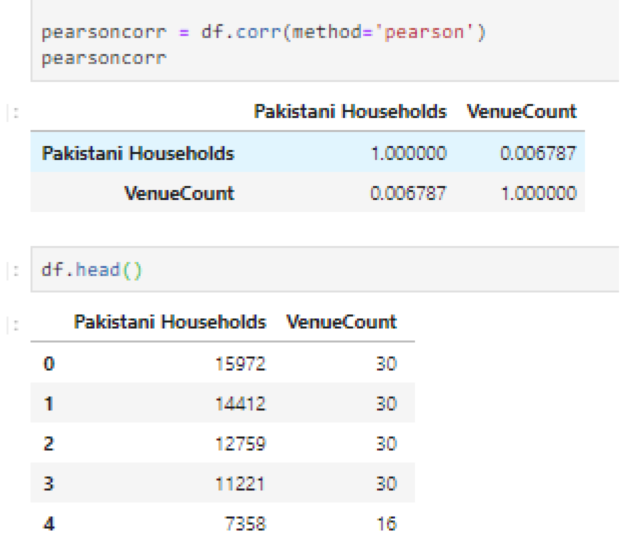
```
                    County  Pakistani Households   Latitude  Longitude  \
0     Queens County, New York                15972  40.652493 -73.791421
1      Kings County, New York                14412  40.645310 -73.955023
2       Cook County, Illinois                12759  41.819738 -87.756525
3        Harris County, Texas                11221  29.811977 -95.374125
4   Fairfax County, Virginia                  7358  38.815636 -77.283685

   VenueCount
0          30
1          30
2          30
3          30
4          16
```

2.3. Feature Selection

For final data frame we dropped Lat, Long columns as we had the venue count with us. Also the county name was dropped because the K-means clustering algorithm that we will opt cannot work with non-numerical values.

Final dataframe with Pearson correlation coefficient was as:



```
pearsoncorr = df.corr(method='pearson')
pearsoncorr
```

	Pakistani Households	VenueCount
Pakistani Households	1.000000	0.006787
VenueCount	0.006787	1.000000

```
df.head()
```

	Pakistani Households	VenueCount
0	15972	30
1	14412	30
2	12759	30
3	11221	30
4	7358	16

3. Methodology

Clustering is one of the methods of unsupervised learning and is a common technique used for statistical data analysis in different fields. Its a Machine Learning technique that involves the grouping of data points. If we have set of data points, we can use a clustering algorithm to segment and classify each data point into a specific group. Data points in the same group ideally should have similar properties and/or features

3.1. K-Means Clustering

There are many models for clustering out there. K-Means model is considered one of the simplest models amongst them. Despite its simplicity, the K-means is vastly used for clustering in many data science applications, especially useful if you need to quickly discover insights from unlabeled data.

Some real-world applications of k-means:

- Customer segmentation
- Understand what the visitors of a website are trying to accomplish
- Pattern recognition
- Machine learning
- Data compression

K-means will partition our households into mutually exclusive groups, for example, into 4 clusters. The households in each cluster are similar to each other in terms of venue count proximity and number of households in that locality. We can create a profile for each group, considering the common characteristics of each cluster.

3.2. Setting up K-Means

The KMeans class has many parameters that can be used, but we will be using these three:

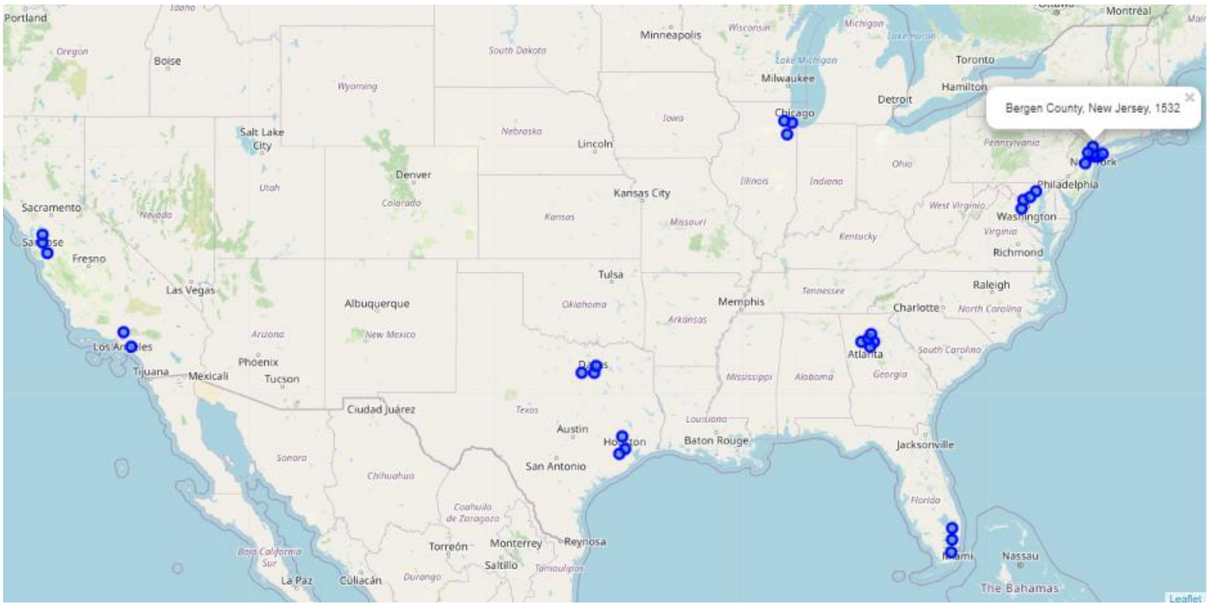
init: Initialization method of the centroids.
Value will be: "k-means++", k-means++: Selects initial cluster centers for k-mean clustering in a smart way to speed up convergence.
n_clusters: The number of clusters to form as well as the number of centroids to generate.
Value will be: 4 for this study
n_init: Number of time the k-means algorithm will be run with different centroid seeds. The final results will be the best output of n_init consecutive runs in terms of inertia.
Value will be: 12

3.3. Normalize the Data

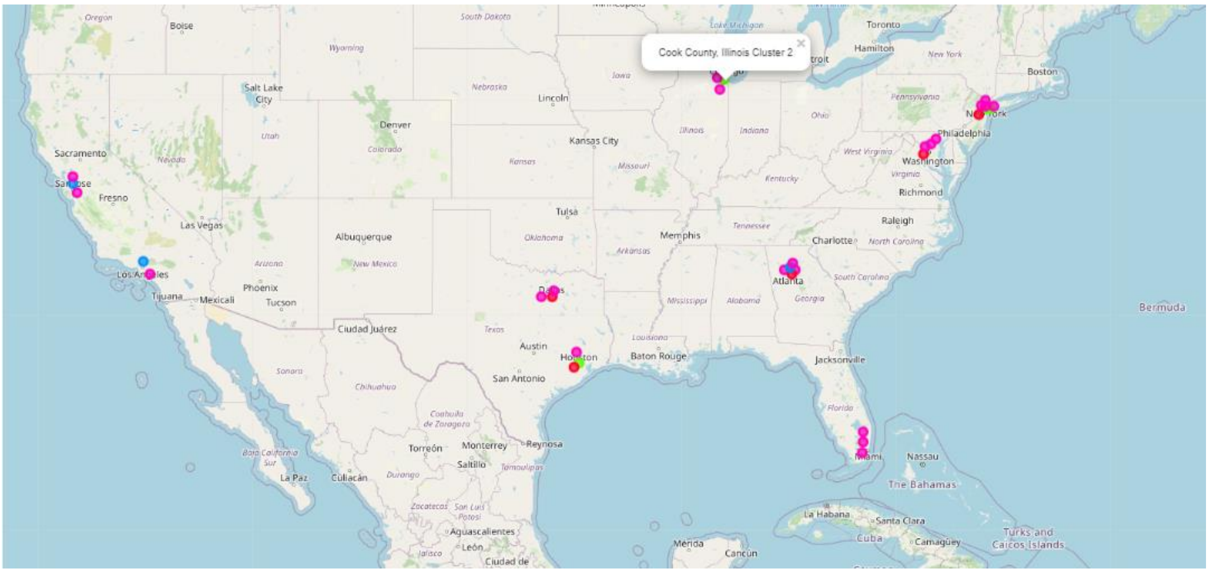
Before running the K-Means algorithm we first Normalize the data. Normalization is a statistical method that helps mathematical-based algorithms interpret features with different magnitudes and distributions equally. We use StandardScaler() to normalize our dataset.

4. Results

Following map shows location of our counties of interest:



After running the clustering algorithm, we can see the counties have been clustered into 4 segments (represented by red – Cluster 2, green – Cluster 3, blue – Cluster 4 and pink – Cluster 1 markers):

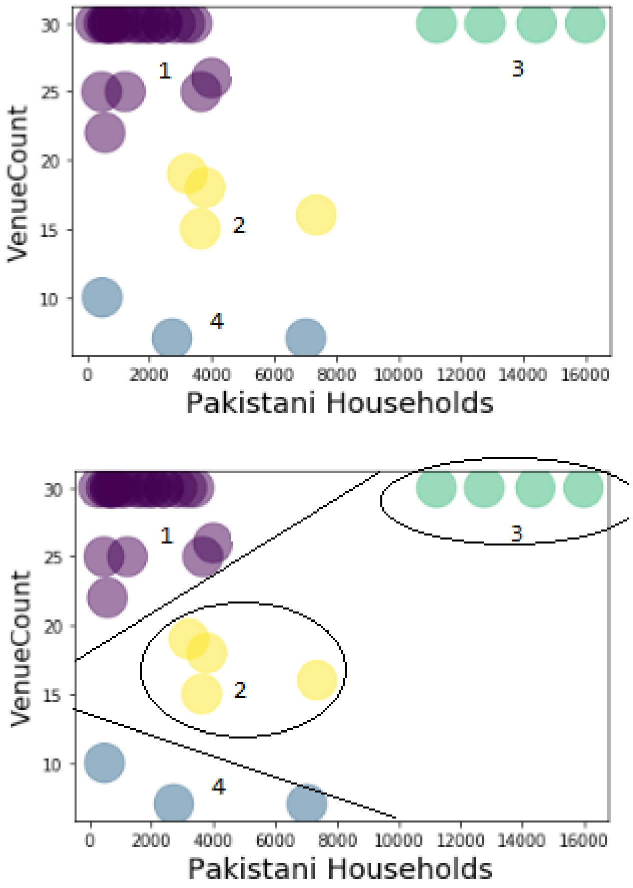


We chose to run the algorithm with cluster size of 4, as per below scatter graph we can see that counties are segmented into 4 segments or clusters which we can name as:

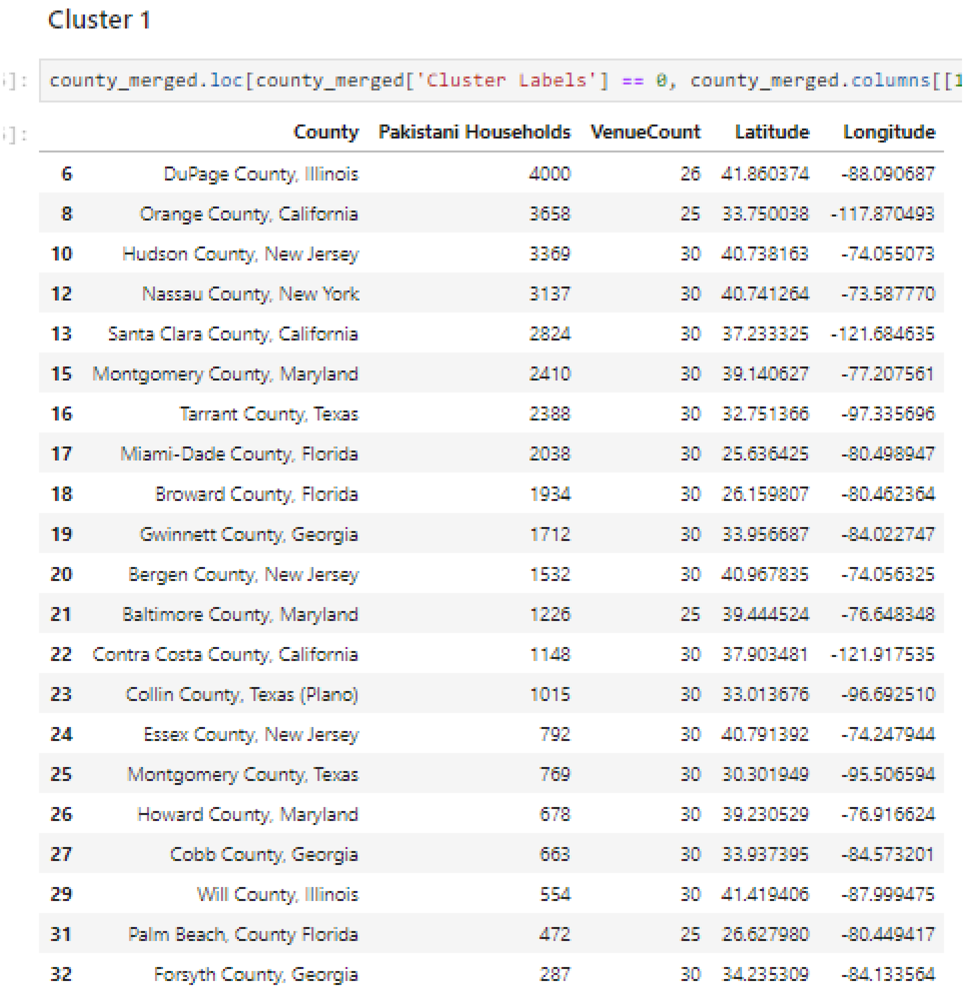
1. **Fewer workplace options and less to average number of households**
   Workplace options less than 10 and Households count below 8000
2. **Average workplace options and number of households**
   Workplace options between 10 and 20, Households count between 4000 to 8000
3. **Average to high workplace options but a low number of households**
   Workplace options higher than 20, Households count less than 4000

4. A High number of workplace options and number of households

Workplace options higher than 30 and Households count between 11000 to 16000



Scatter Plot showing counties distribution in 4 clusters



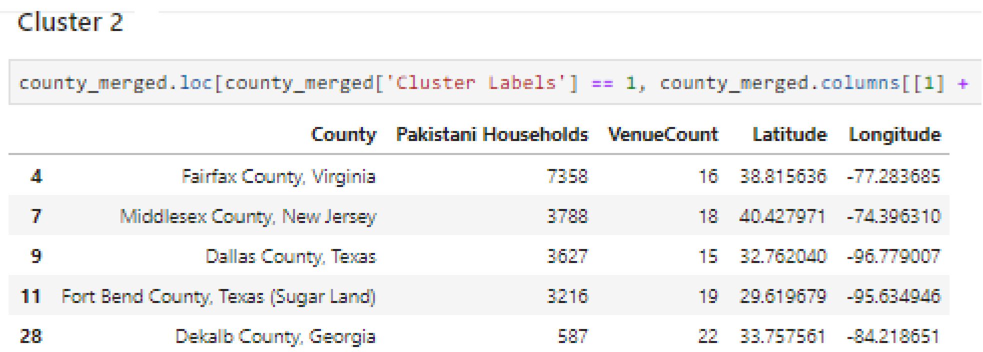
Cluster 1

```
]: county_merged.loc[county_merged['Cluster Labels'] == 0, county_merged.columns[[1
```

]:

	County	Pakistani Households	VenueCount	Latitude	Longitude
6	DuPage County, Illinois	4000	26	41.860374	-88.090687
8	Orange County, California	3658	25	33.750038	-117.870493
10	Hudson County, New Jersey	3369	30	40.738163	-74.055073
12	Nassau County, New York	3137	30	40.741264	-73.587770
13	Santa Clara County, California	2824	30	37.233325	-121.684635
15	Montgomery County, Maryland	2410	30	39.140627	-77.207561
16	Tarrant County, Texas	2388	30	32.751366	-97.335696
17	Miami-Dade County, Florida	2038	30	25.636425	-80.498947
18	Broward County, Florida	1934	30	26.159807	-80.462364
19	Gwinnett County, Georgia	1712	30	33.956687	-84.022747
20	Bergen County, New Jersey	1532	30	40.967835	-74.056325
21	Baltimore County, Maryland	1226	25	39.444524	-76.648348
22	Contra Costa County, California	1148	30	37.903481	-121.917535
23	Collin County, Texas (Plano)	1015	30	33.013676	-96.692510
24	Essex County, New Jersey	792	30	40.791392	-74.247944
25	Montgomery County, Texas	769	30	30.301949	-95.506594
26	Howard County, Maryland	678	30	39.230529	-76.916624
27	Cobb County, Georgia	663	30	33.937395	-84.573201
29	Will County, Illinois	554	30	41.419406	-87.999475
31	Palm Beach, County Florida	472	25	26.627980	-80.449417
32	Forsyth County, Georgia	287	30	34.235309	-84.133564

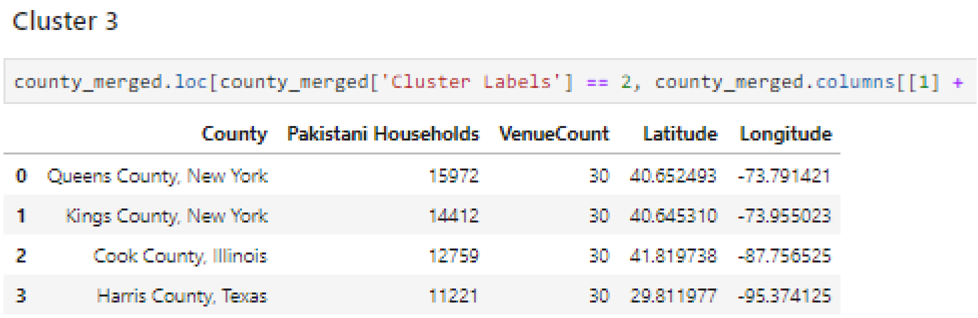
List of Counties in Cluster 1



Cluster 2

```
county_merged.loc[county_merged['Cluster Labels'] == 1, county_merged.columns[[1] +
```

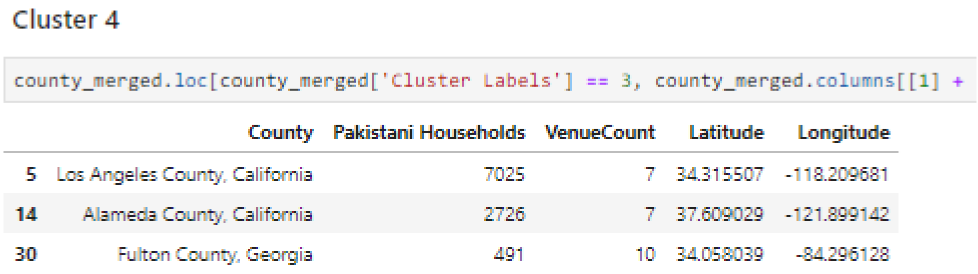
	County	Pakistani Households	VenueCount	Latitude	Longitude
4	Fairfax County, Virginia	7358	16	38.815636	-77.283685
7	Middlesex County, New Jersey	3788	18	40.427971	-74.396310
9	Dallas County, Texas	3627	15	32.762040	-96.779007
11	Fort Bend County, Texas (Sugar Land)	3216	19	29.619679	-95.634946
28	Dekalb County, Georgia	587	22	33.757561	-84.218651



Cluster 3

```
county_merged.loc[county_merged['Cluster Labels'] == 2, county_merged.columns[[1] +
```

	County	Pakistani Households	VenueCount	Latitude	Longitude
0	Queens County, New York	15972	30	40.652493	-73.791421
1	Kings County, New York	14412	30	40.645310	-73.955023
2	Cook County, Illinois	12759	30	41.819738	-87.756525
3	Harris County, Texas	11221	30	29.811977	-95.374125



Cluster 4

```
county_merged.loc[county_merged['Cluster Labels'] == 3, county_merged.columns[[1] +
```

	County	Pakistani Households	VenueCount	Latitude	Longitude
5	Los Angeles County, California	7025	7	34.315507	-118.209681
14	Alameda County, California	2726	7	37.609029	-121.899142
30	Fulton County, Georgia	491	10	34.058039	-84.296128

List of Counties in Cluster 2, 3, 4

5. Discussion and Recommendations

At data gathering stage different data sources can be explored to get more recent and detailed data sets. We can also segment based on neighborhoods or state level. The clustering algorithm can be further improved by taking into consideration other factors such as

- County Population
- Concentration of Pakistanis
- Distance to workplace, school/college/university for a user's children
- Climate severity
- price of property or rent
- proximity and availability of amenities

Also, Foursquare venue search API is returning currently 30 venues at max and a solution or alternate for this can be used.

The solution can be made interactive for user inputs and more visual graphs can be included.

6. Conclusion

In this study, I utilized Machine Learning’s K-Means clustering algorithm to group available information regarding foreigner population distribution in US into distinct groups or clusters based on household count, workplace count and proximity to workplace. The visual graphs give a user quick insight as to which all counties exist in the same category and what all categories or segments exist out there. This can help them make an informed decision while choosing a county for abode.