Where Should You Live?

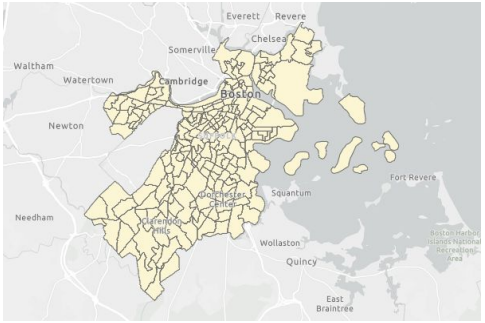
Relocation Recommendation with Neighborhood Clustering

Nhat Pham, Kahan Sheth, Vignesh Sankar

Value Proposition

- As Zillow, we seek to enhance user experience by providing precise, personalized relocation recommendations
- Navigating multiple factors *(eg. cost, employment, social characteristics)* can be cumbersome using traditional search methods
- We use clustering algorithms to identify *neighborhoods* with shared traits, characterize familiar places with these results, and create a recommendation tool to suggest *places* based on user preferences.

Data

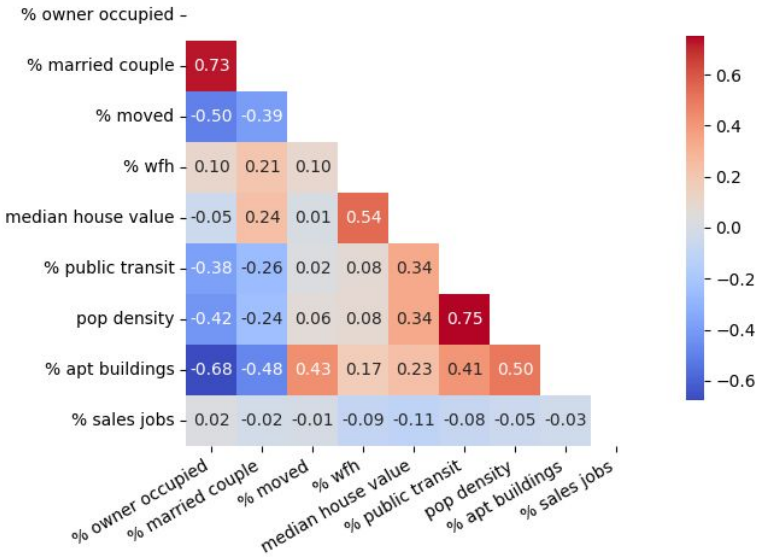


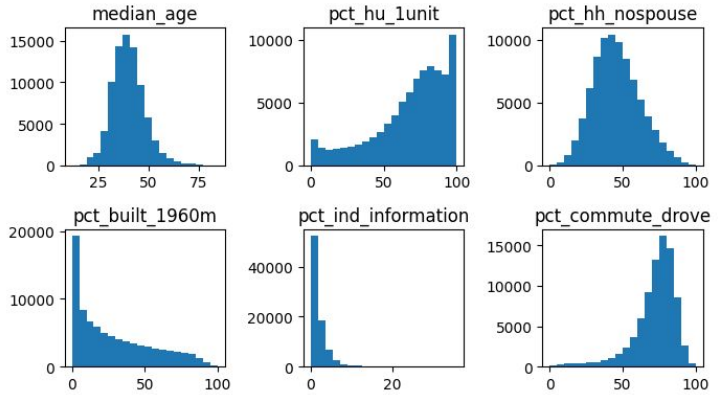
- **Census tracts**: small areas with similar population, offering a more comparable unit for analysis than larger, uneven regions like counties or zip codes
- Sources: multiple US Census Bureau datasets, covering ~**83,000 tracts**
- **67 features**:
  - **Demographic**: eg. age, race
  - **Economic**: eg. industry, income, cost of living
  - **Housing**: eg. unit structure, tenure, occupancy type
  - **Social**: eg. household type, education, mobility, politics
  - **Geographic**: eg. land use, population density

EDA

- Some features show strong linear correlation, indicating potential for linear projection

- Feature vary in distribution; most are percentages (0-100) while others differ in range significantly

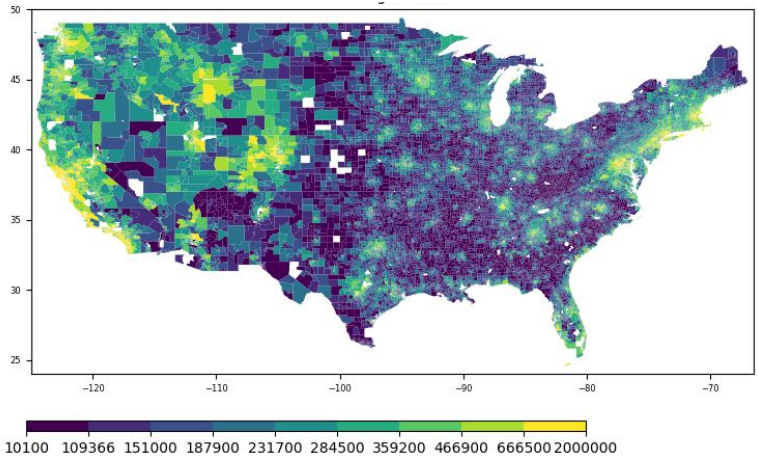




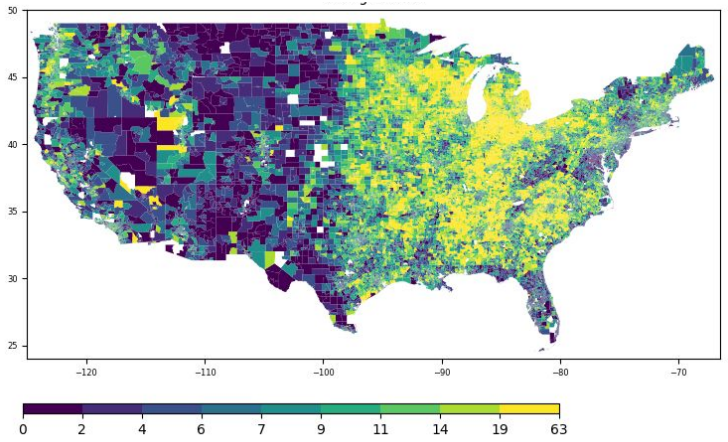
EDA

Some features show regional patterns while others dispersed across the country

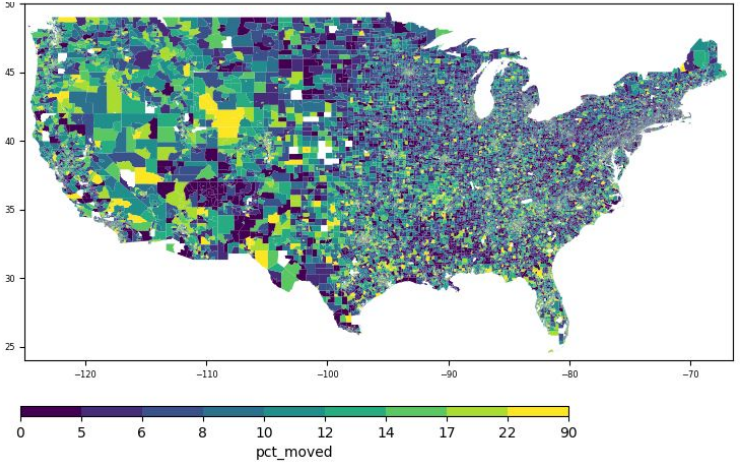
Median house value



Percentage work in manufacturing

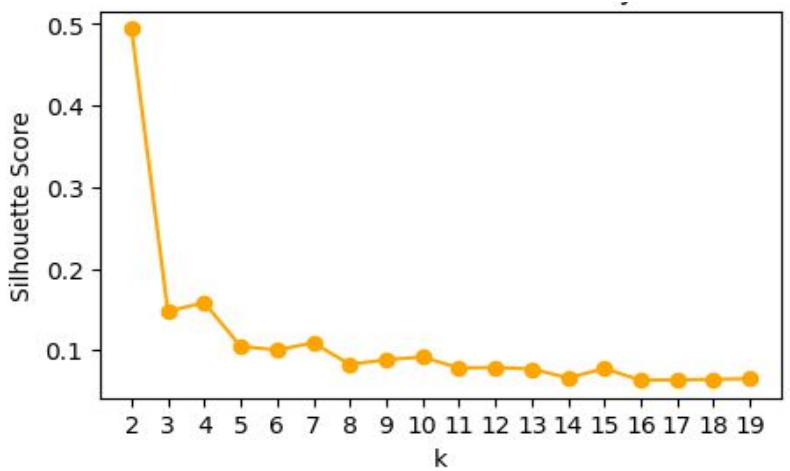


Mobility rate



Baseline Model

- KMeans clustering on the complete dataset yielded unsatisfactory results
  - Low silhouette scores indicate poor cluster quality
  - Imbalanced cluster sizes
- We will apply dimensionality reduction to improve model effectiveness

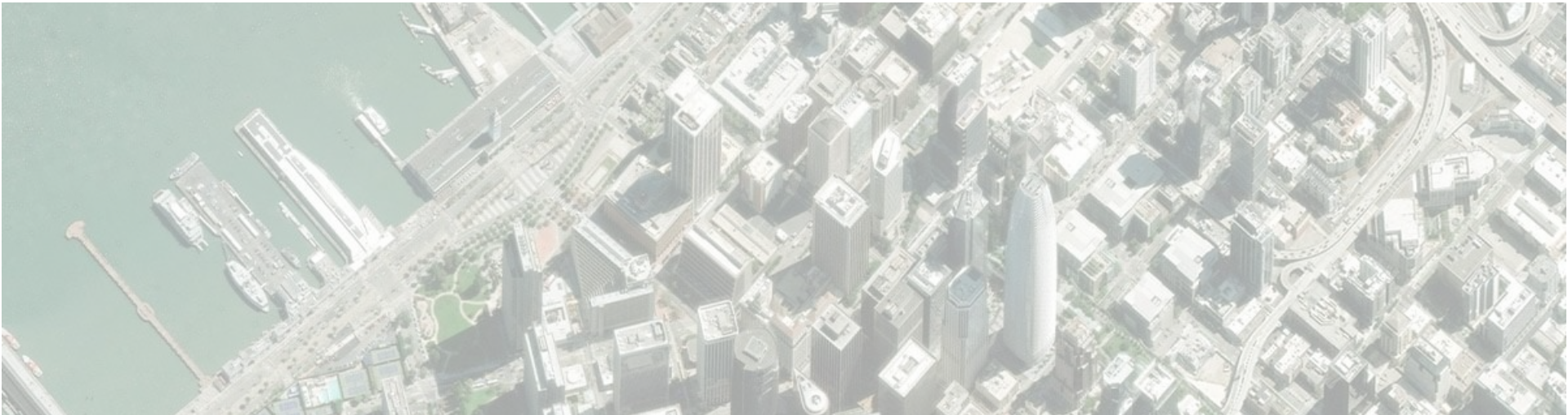


Silhouette score, baseline KMeans

Cluster	Count
4	19478
12	15683
1	10136
0	8408
2	8062
11	5967
7	4051
8	3748
6	2901
9	1391
10	1316
3	1303
5	525

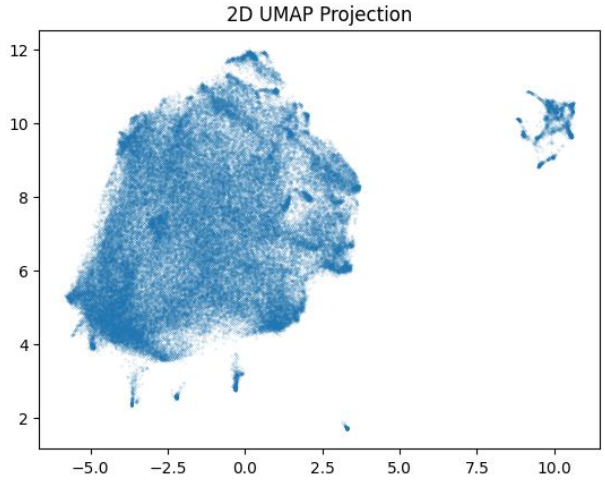
Cluster size, baseline KMeans

Dimensionality Reduction



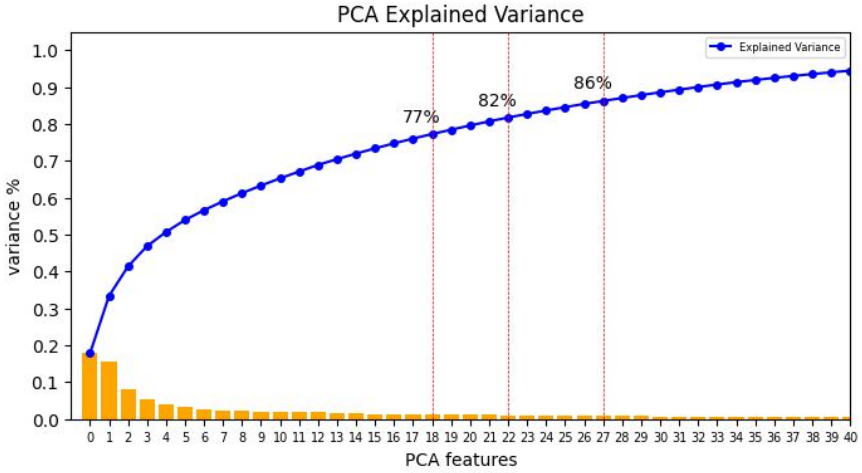
t-SNE, UMAP

- Visual inspection & testing to select parameters values
- Tested clustering with UMAP 3, 12, 18 dimensions. 12D yielded most meaningful clusters.

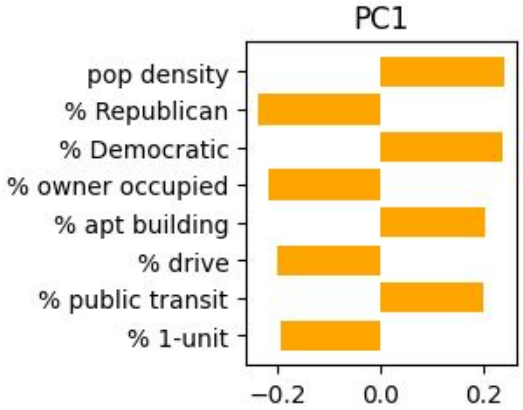


PCA

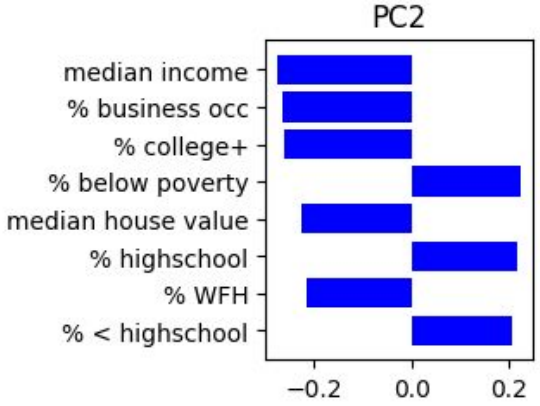
- Applied Power Transformer on skewed features, then standardized all features
- Tested clustering with 18, 22, 27 PCs. Selected **18 PC**s to balance model effectiveness and variance explained



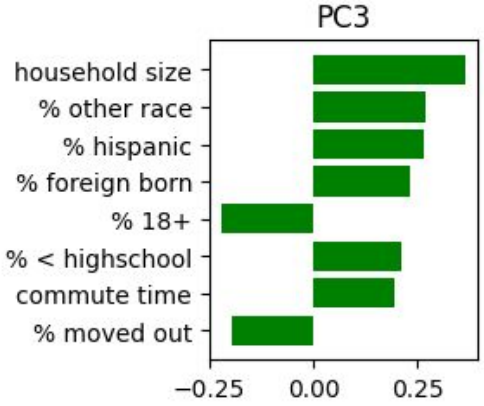
PC Loadings



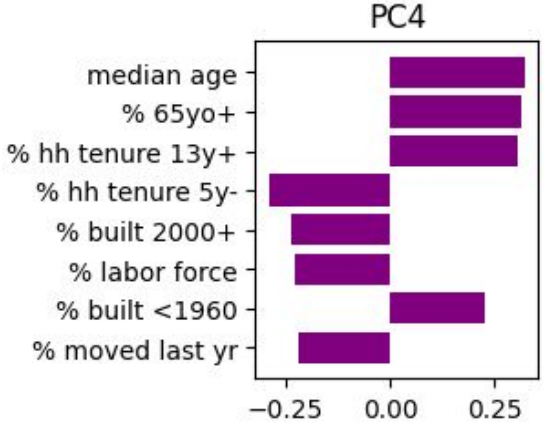
urban



economic difficulty



family

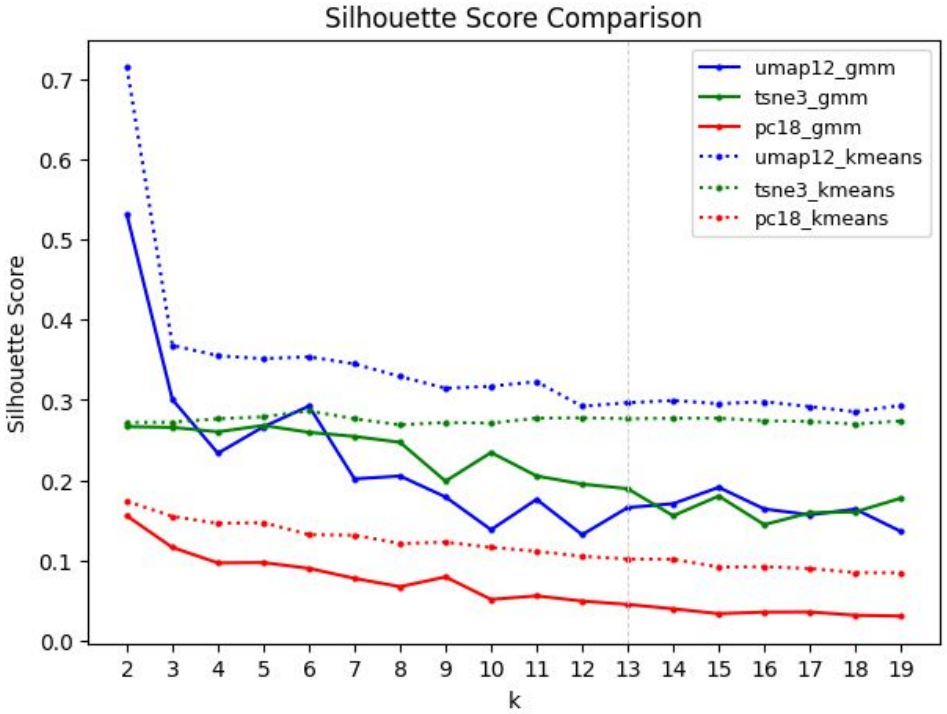


seniority

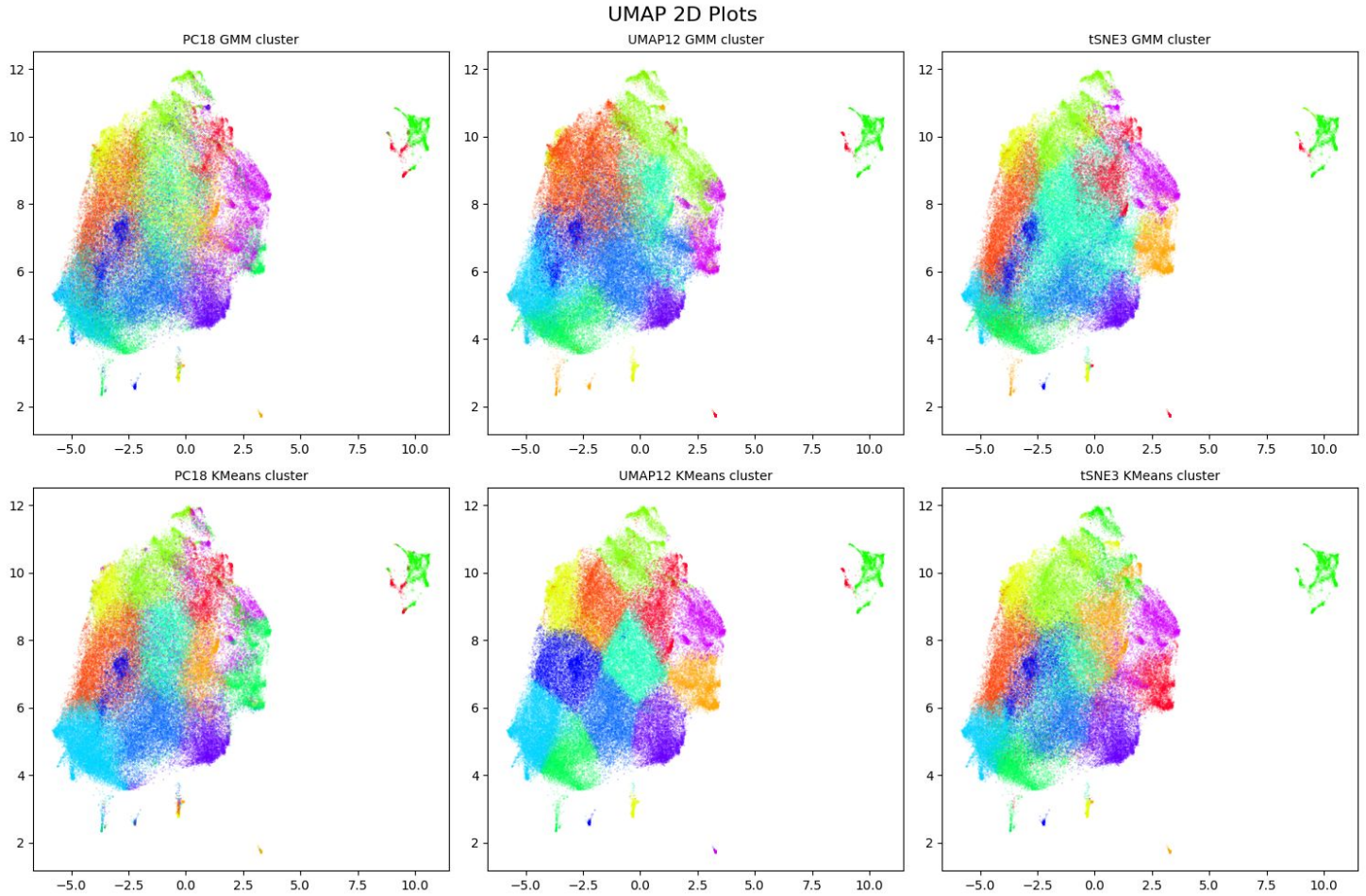
Neighborhood Clustering

Model Selection

- Selected **GMM** and **KMeans**
  - HDBSCAN produced 1 large cluster, mem
    issue with Agglomerative Hierarchical
- **13 clusters** provided stable and meaningful groupings, aligned with industry insight

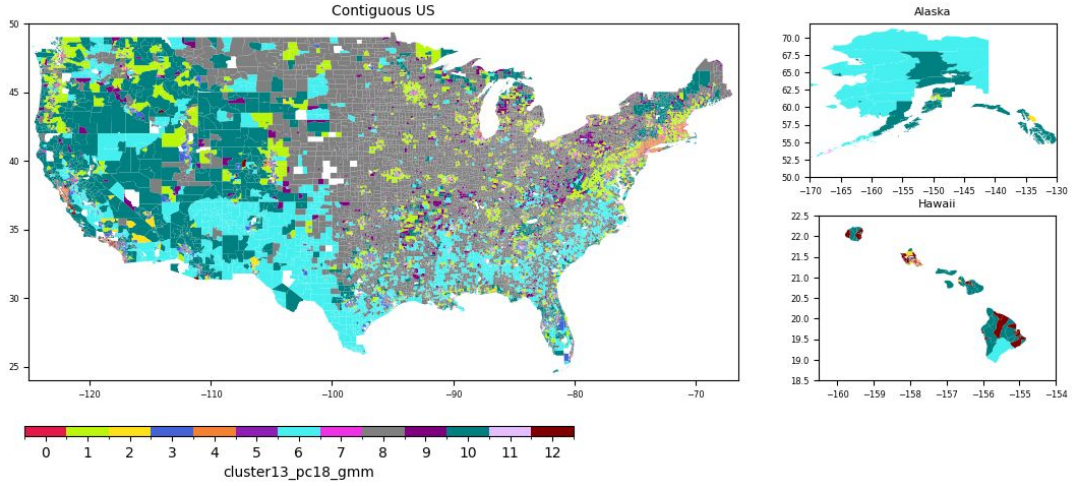


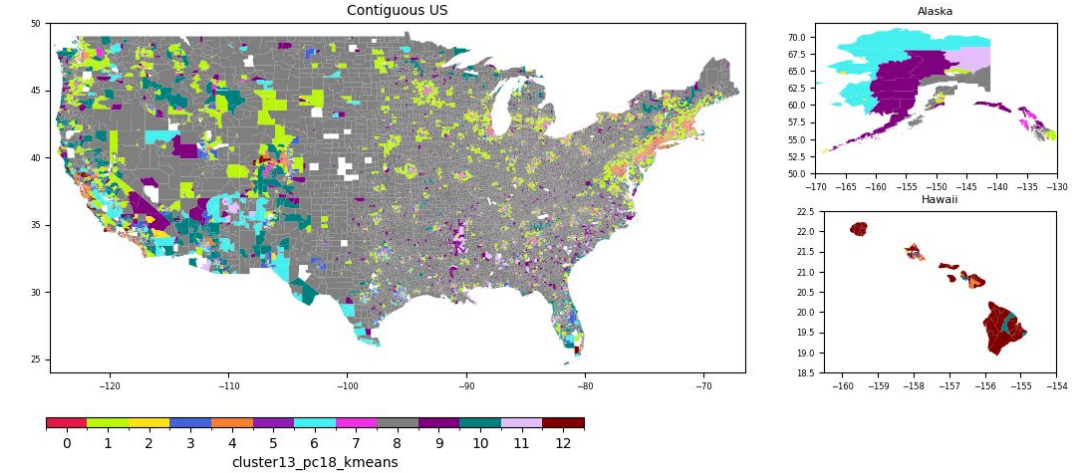
Model Comparison



PCA + GMM vs KMeans

- Cluster results are generally similar
- KMeans broadly groups rural areas into one cluster
- GMM found subtle distinctions within similar density region eg. diversity levels in rural areas
- Similar findings on models with UMAP and tSNE
- **Select GMM over KMeans**





PCA/tSNE/UMAP+ GMM

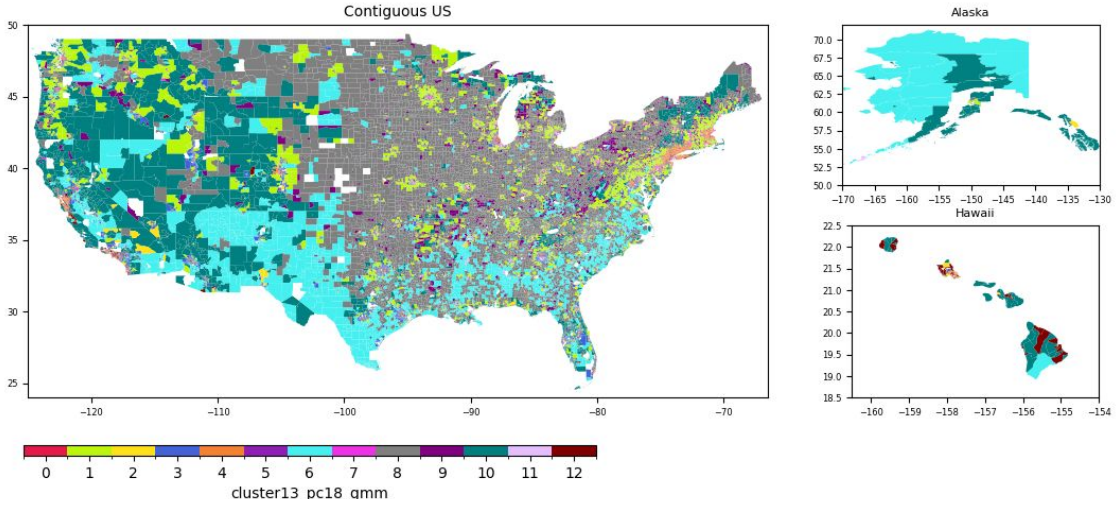
tSNE

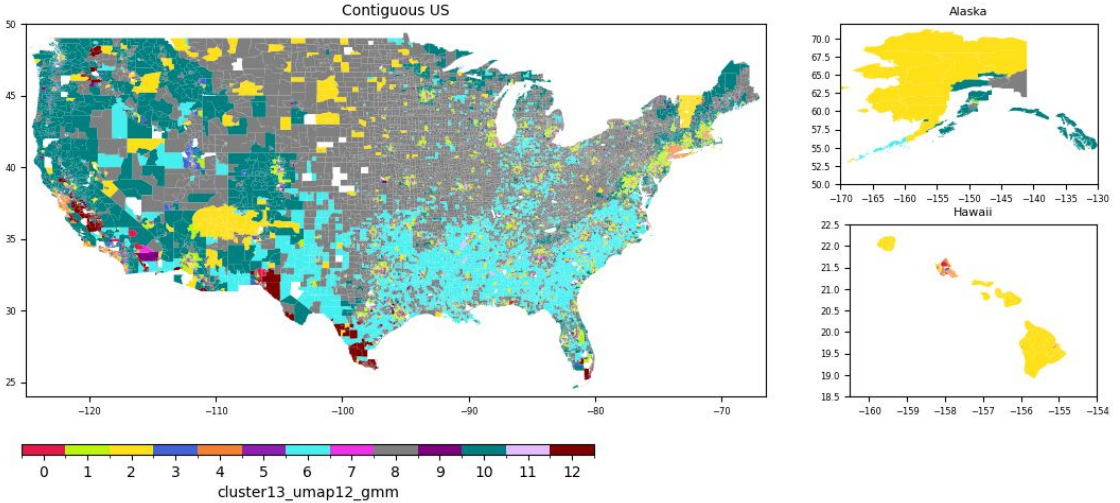
- Creates smaller, ‘cleaner’ clusters
  eg. Separates oil/gas areas (AK, NM, TX) from a bigger ‘construction’ cluster
- Forms a big indistinct cluster of 20k tracts
- Groups Hawaii with metro areas based on diversity but missed density differences

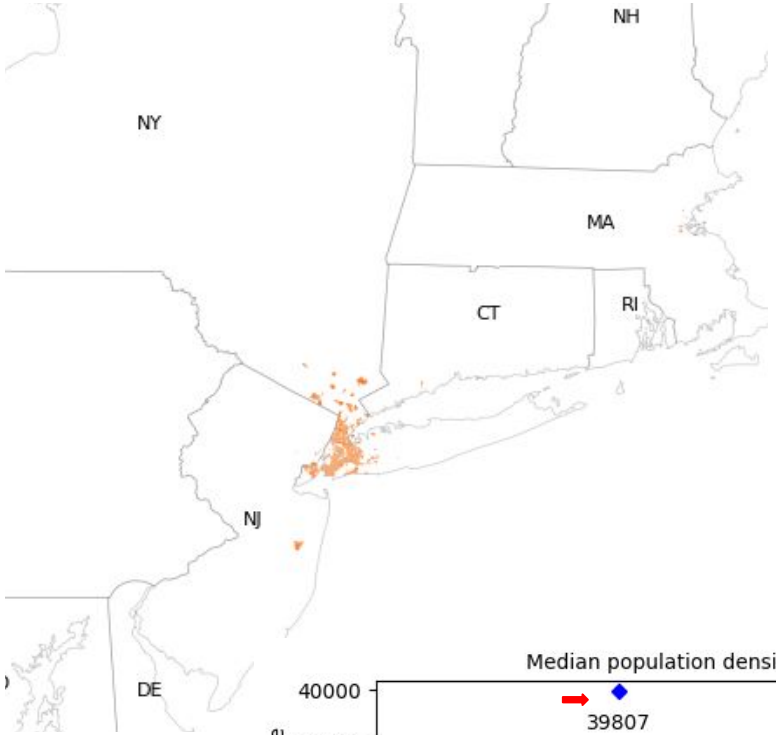
UMAP

- Forms a big indistinct cluster of 19k tracts
- Combines areas with similar density yet diverse other traits (HI, AK, NM, SD)

- **Selected GMM + PCA** for its balanced differentiation

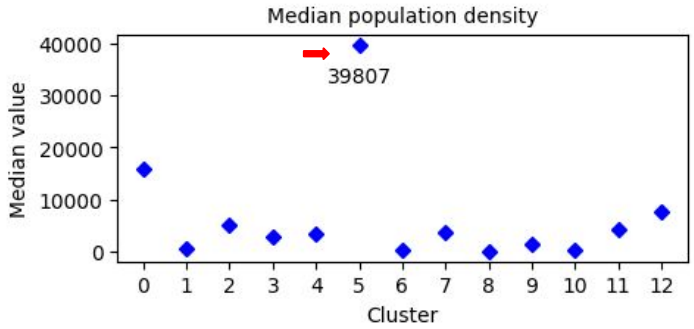


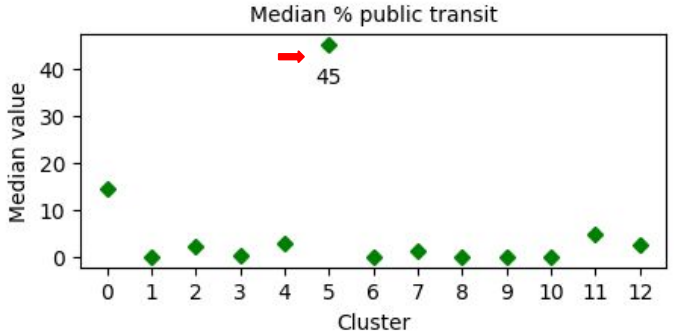




Downtown Melting Pot

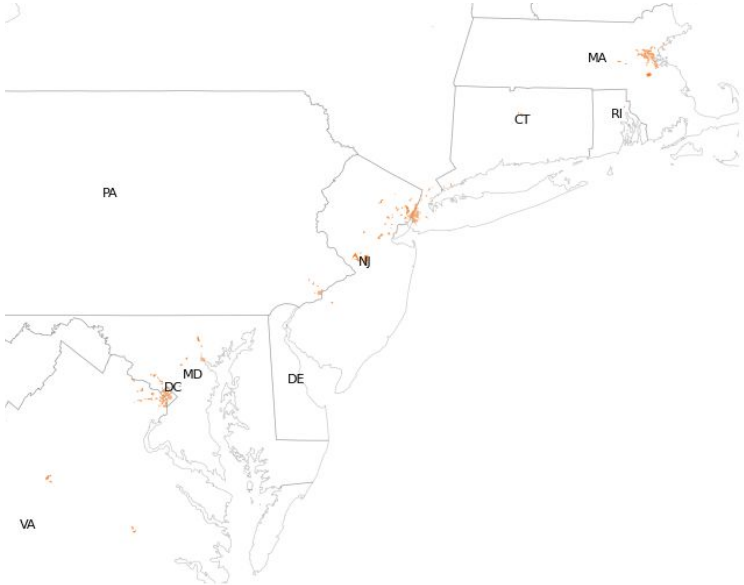
- Most diverse, densely populated
- Extensive public transit, long commute
- Old rental housing, high living cost
- Many work in education, healthcare, service
- *eg. NYC boroughs, Downtown SF, Mt. Vernon city, Yonkers city (NY)...*

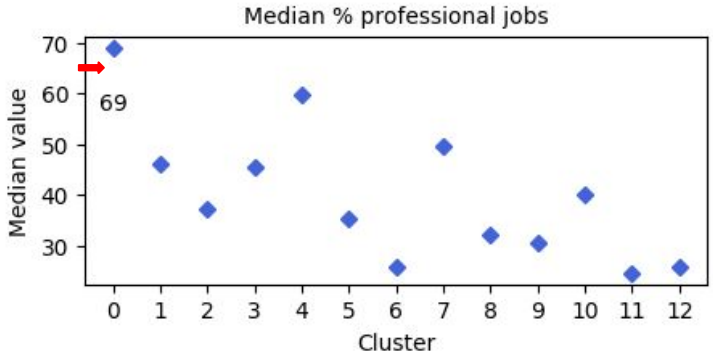


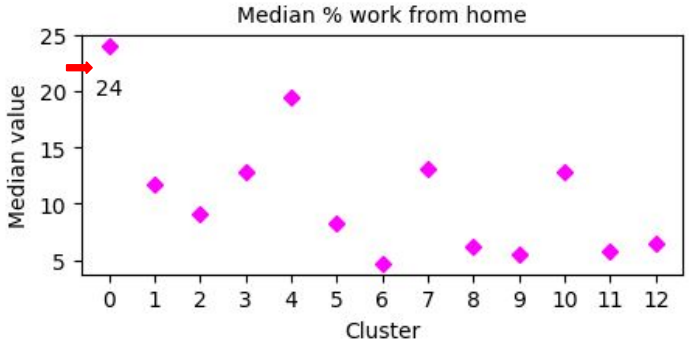


Enterprising Professionals

- Young, educated professionals in STEM. High remote work rate
- Single households, rental properties
- High income, high living cost
- Notable Asian & foreign-born residents
- *eg. Chicago, LA, DC, Boston, Seattle, San Jose…*

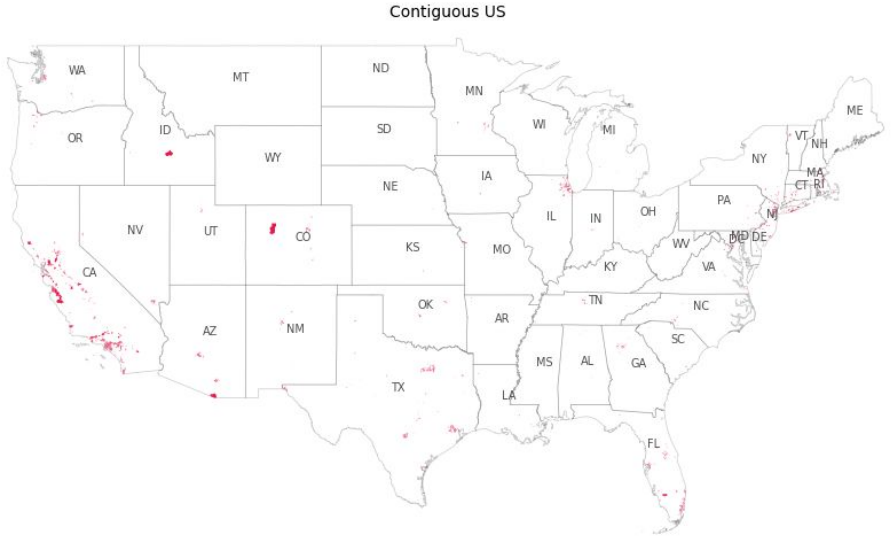


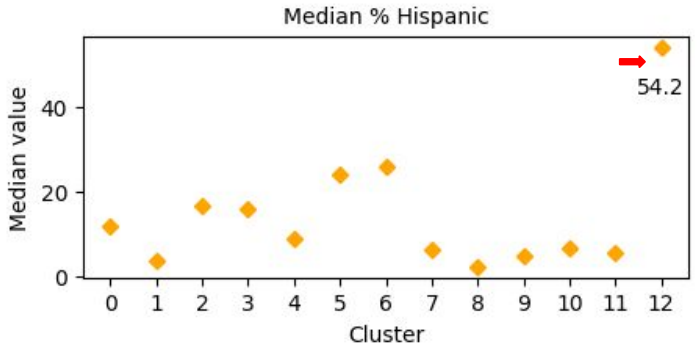




Urban Hispanic Families

- Urban periphery of large metro area, primarily West Coast
- Hispanic & foreign-born residents, large household size
- Commute by car, notably carpooling
- Many work in construction, service, retail
- High % below high school education
- *LA, Houston, Chicago, Orange (CA), Phoenix, San Diego,...*

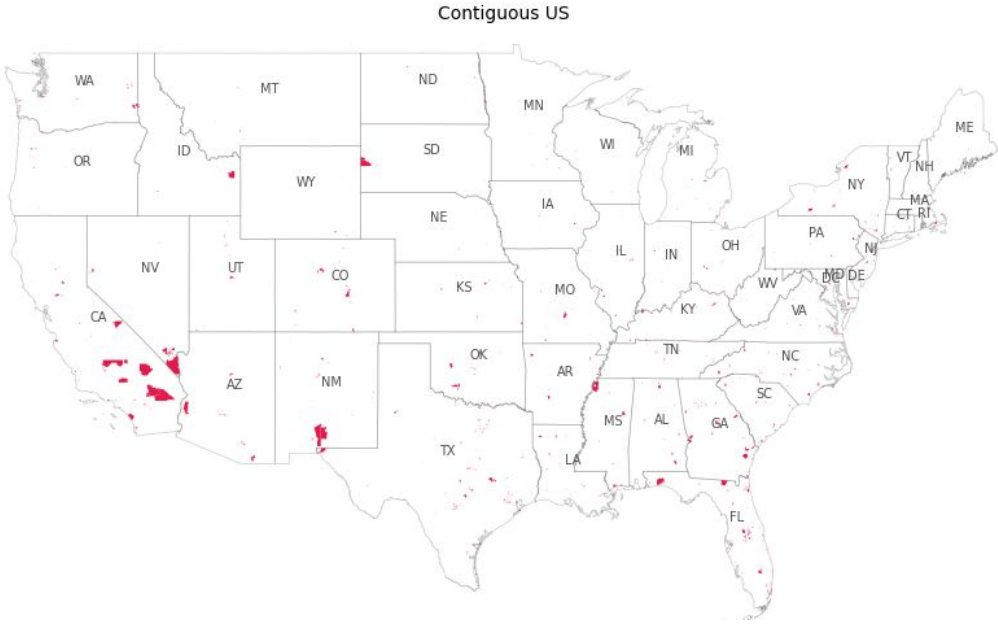


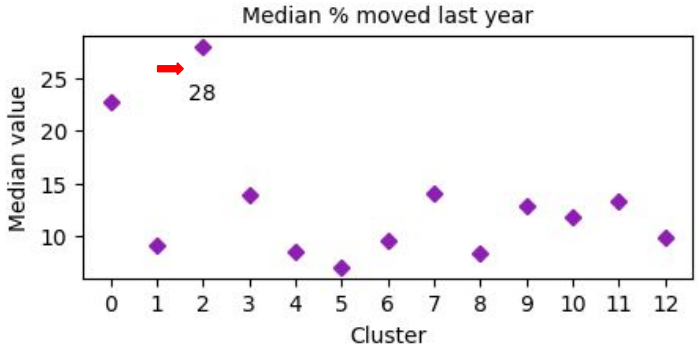


Midtown Singles

- Youngest, most mobile population
- Mainly new, rental housing, single households
- Lower than average living cost, yet high rent burden
- Many work in service, arts, education, healthcare
- Low income, high poverty

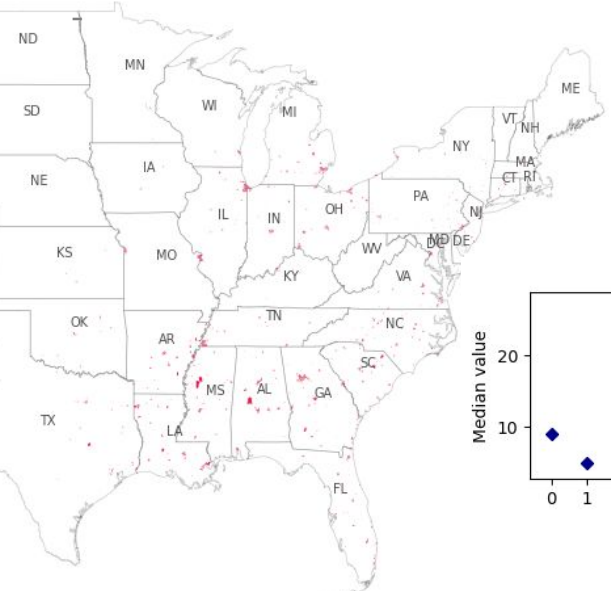
eg. Las Vegas, Houston, Miami, LA, Honolulu, Austin, Atlanta

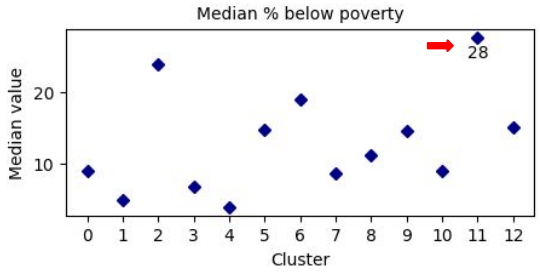




Modest Income Homes

- High % Black residents, single households, historic homes, low house values
- High poverty rate, low income, rental burden
- Many work in transportation, sales, service
- *eg. Detroit, Philly, Baltimore, Cleveland, Milwaukee, New Orleans*

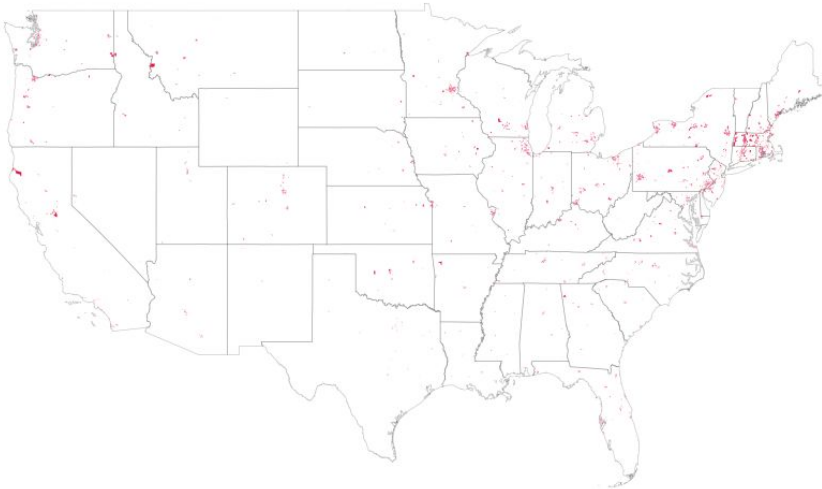




Emerald City

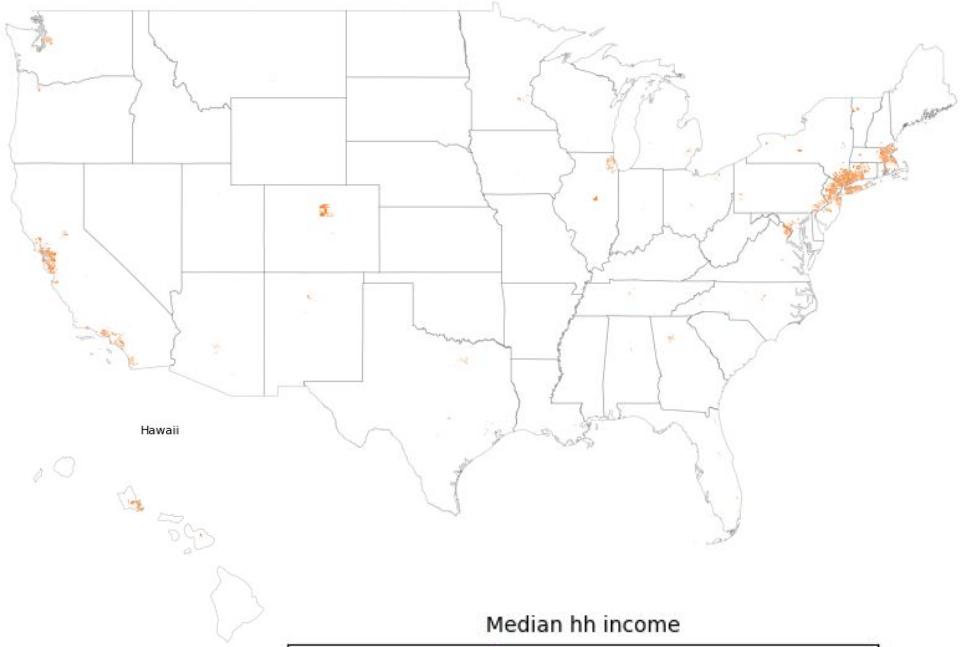
- Low density, old neighborhoods in urban areas
- Many have college degree, professional jobs, work from home
- Above median income ($80k)

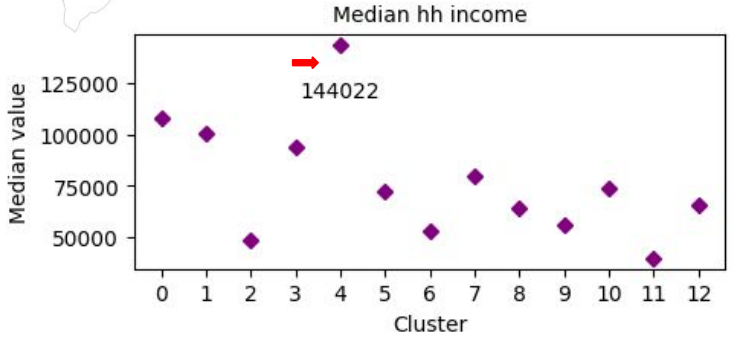
eg. East Portland, Seattle, Minneapolis, Philly, Salt Lake City, Sacramento



Affluent Pleasantville

- Older families in suburban peripheries of large metro areas, especially NY and CA
- High homeownership rate, low mobility
- Many work in finance/tech
- Highest median income
- *eg. San Jose, San Diego, Hempstead town, Brookhaven town (NY), Contra Costa (CA)*

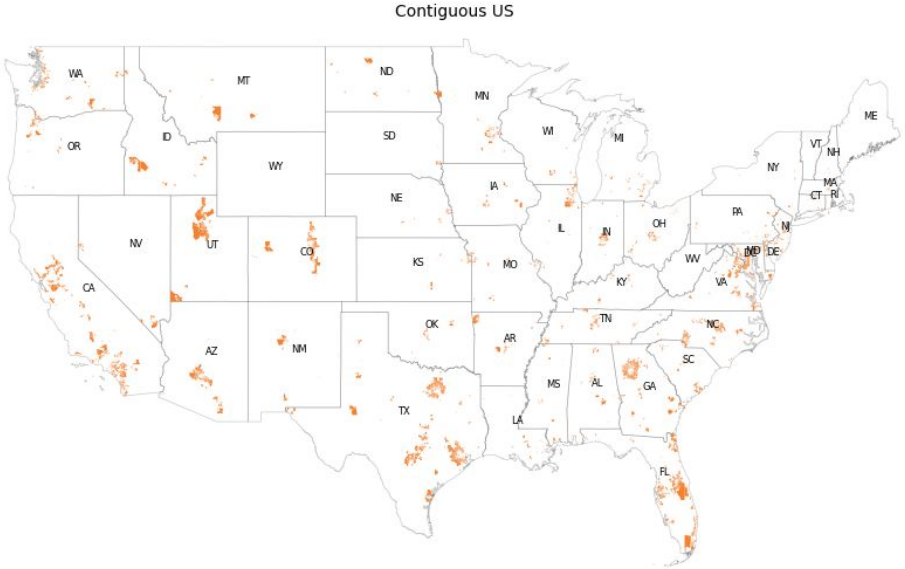




Up and Coming Families

- Young professionals with families in the suburbs with the newest housing
- High homeownership, mostly single family units, >average house value
- High education level & median income $93k
- Car dependent

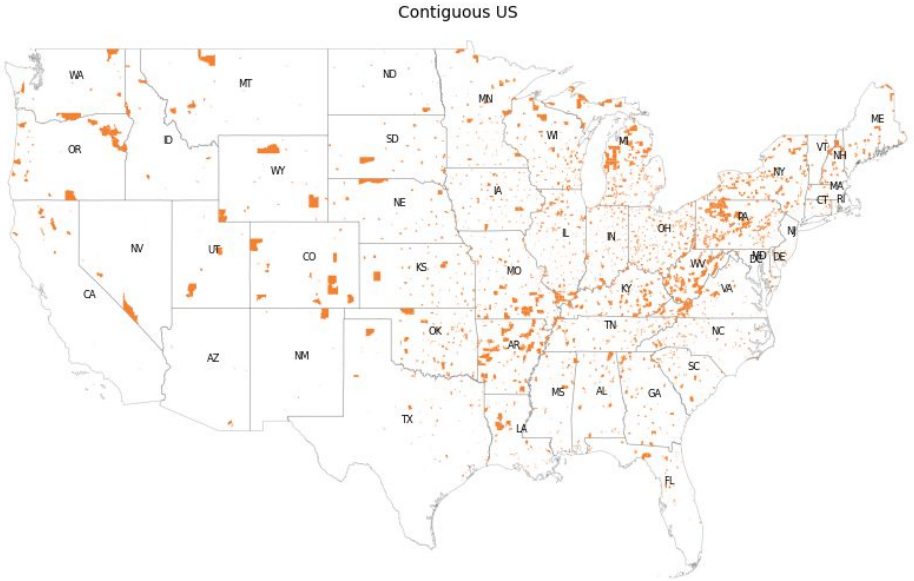
eg. Phoenix, Houston, Las Vegas, Charlotte (NC), Dallas, Salt Lake City

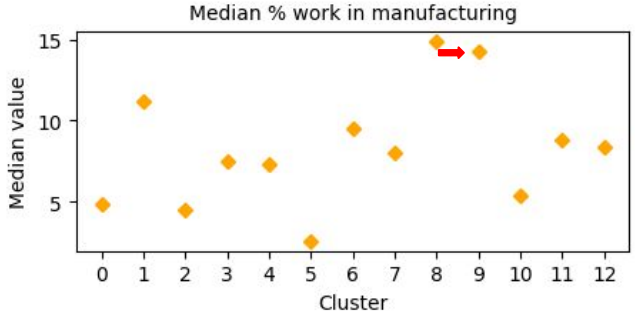


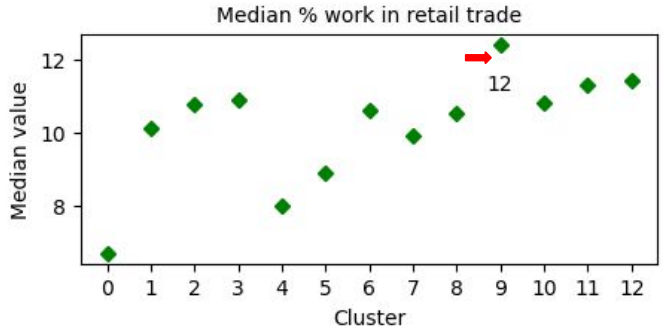
Affordable Industrial Towns

- Long-tenured residents working in manufacturing, retail trade, transportation
- Older, affordable housing; mostly single-units
- High % high school graduates

eg. Toledo (OH), Wichita (KS), Tulsa (OK), Warren (MI), Columbus (OH)



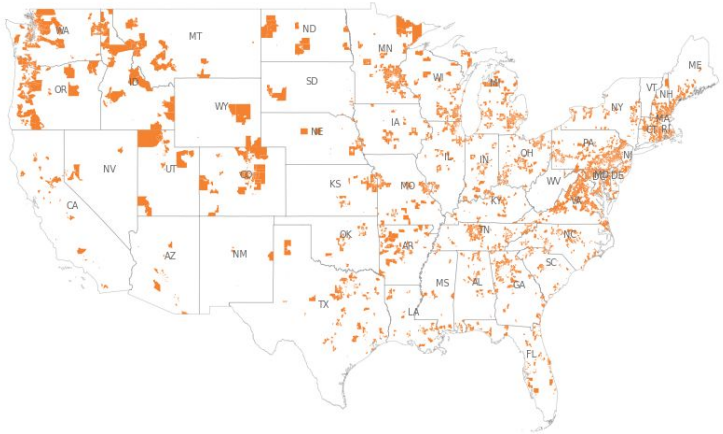




Savvy Suburbanites

- Conservative, affluent older families
- High home & vehicle ownership
- Many work in professional & manufacturing, high income ($100k)

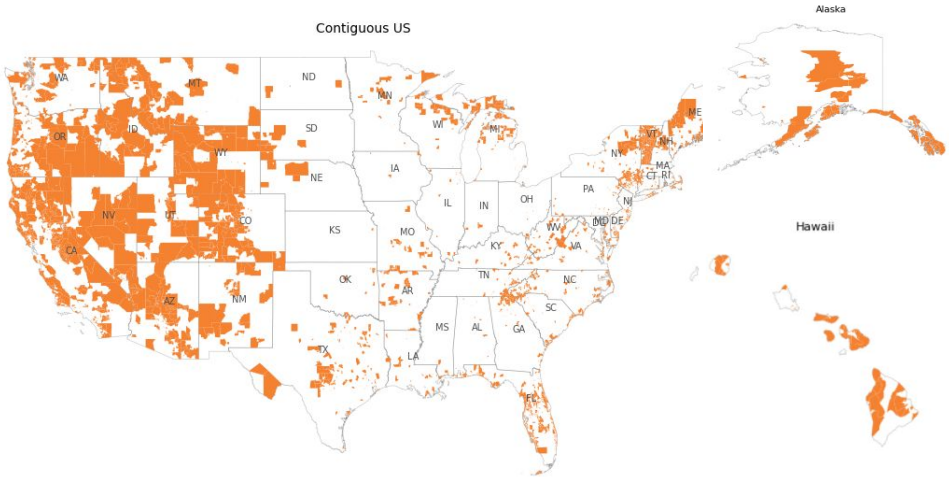
eg. Tulsa, Shelby (AL), Boise City, Lexington (SC), Chesaspeake (VA)



Senior Communities

- Senior residents (median age 53.2)
- High % retirement & social security incomes
- Mostly single-units, average housing cost

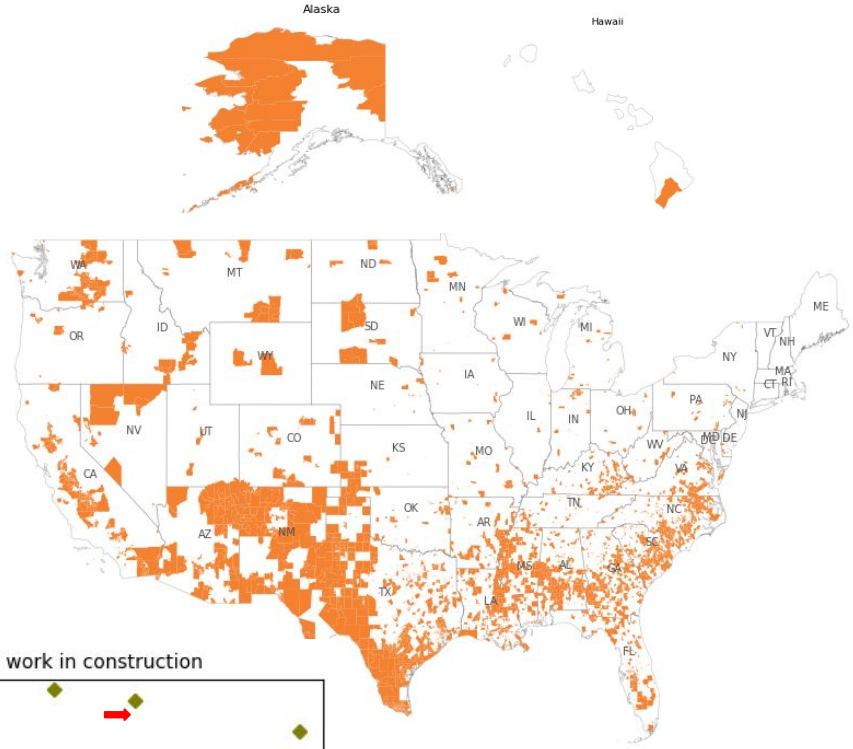
eg. Naples, Clearwater, Palm Beach, Sarasota, Manatee (FL)

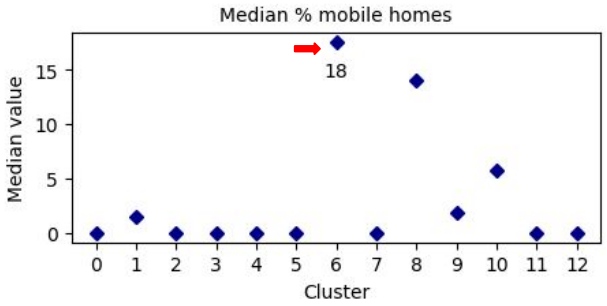


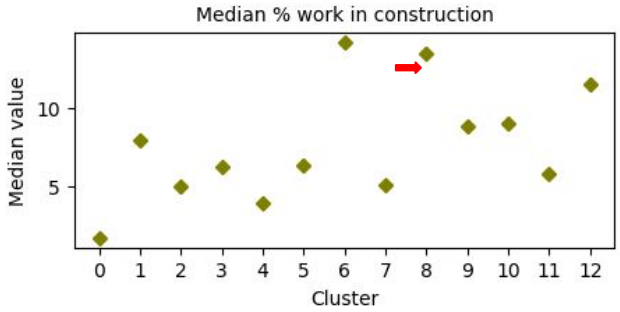
Economic BedRock

- Diverse rural population, mainly in the South
- High homeownership rate, high % mobile homes
- Many work in construction, agriculture, transportation
- Below average income, high poverty rate

eg. Hidalgo, Cameron, El Paso, Webb (TX), San Bernardino (CA)

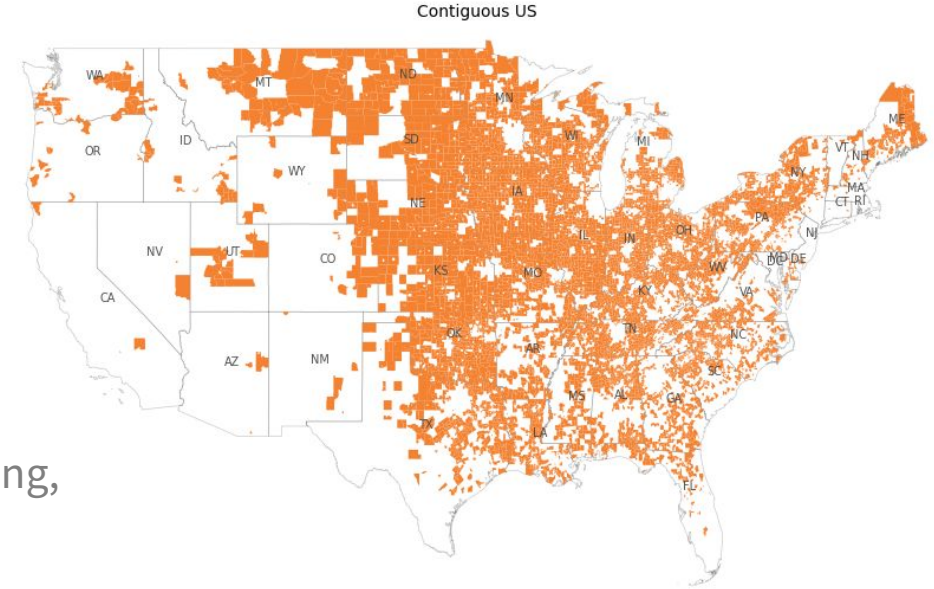


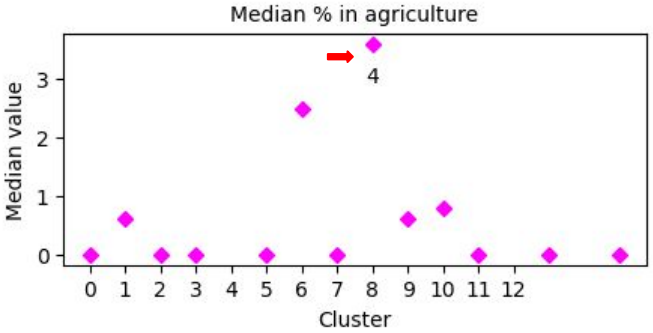


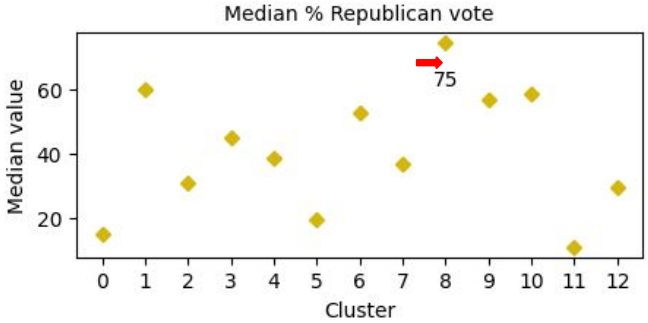


Prairie Living

- Most rural areas, mainly in Midwest
- Families that own single-unit homes, and many vehicles
- Low rent & house values
- Many work in agriculture and manufacturing, low financial housing burden







Location Recommendation

City-Based Recommendation

- **Percentage Calculation:** Determine the percentage of each city's presence in each cluster.
- **Example:** If Jersey City has 3 instances in the data, and 2 fall under cluster 5 and 1 falls under cluster 2, then Jersey City is 66% in cluster 5 and 33% in cluster 2.
- **User Input:** Based on the user's input city, query features are obtained.
- **K-Nearest Neighbor Algorithm:** Use this algorithm to find the 5 nearest neighbors to the user's input city and recommend them to the user.

```
user_input = "Boston city, Suffolk MA"
k_nearest_neighbors = find_k_nearest_neighbors(pivot_table, user_input, k=5)

print(f"Recommended Places to move similar to '{user_input}':")
for i, mcd_name in enumerate(k_nearest_neighbors, start=1):
    print(f"{i}. {mcd_name}")
```

```
Recommended Places to move similar to 'Boston city, Suffolk MA':
1. Jersey City city, Hudson NJ
2. Alexandria city, Alexandria city VA
3. Watertown Town city, Middlesex MA
4. Downtown-Northeast Neighborhoods-Treasure Island CCD, San Francisco CA
5. Manhattan borough, New York NY
```

Tag-Based Recommendation

- **Tag Assignment:** Each cluster is assigned relevant tags that best describe its characteristics.
- **Feature Mapping:** Each place is represented by these tags. For instance, if Boston is 80% within a cluster which is tagged as “Public transit & walkable”, Boston will have an 80% feature value for this tag.
- **User Input:** Based on the tags selected by the user a query feature is created.
- **K-Nearest Neighbor Algorithm:** Apply this algorithm to identify the top 5 places that closely match the user’s preferences.

```python
#User inputs
user_inputs = ["Public transit & walkable"]

k_nearest_neighbors = find_k_nearest_neighbors_third_approach(tags_df, query_vector, k=5)

names_list = []
print(f"Recommendations based on the user preferences of", user_inputs)
for i, mcd_name in enumerate(k_nearest_neighbors, start=1):
    names_list.append(mcd_name)
    print(f"{i}. {mcd_name}")
```

```
Recommendations based on the user preferences of ['Public transit & walkable']
1. Lakewood township, Ocean NJ
2. White Plains city, Westchester NY
3. Ramapo town, Rockland NY
4. New Rochelle city, Westchester NY
5. Yonkers city, Westchester NY
```

```python
#User inputs
user_inputs = ["Public transit & walkable", "Affordable housing", "White collar jobs"]

k_nearest_neighbors = find_k_nearest_neighbors_third_approach(tags_df, query_vector, k=5)

names_list = []
print(f"Recommendations based on the user preferences of", user_inputs)
for i, mcd_name in enumerate(k_nearest_neighbors, start=1):
    names_list.append(mcd_name)
    print(f"{i}. {mcd_name}")
```

```
Recommendations based on the user preferences of ['Public transit & walkable', 'Affordable housing', 'White collar jobs']
1. District 19, Davidson TN
2. Evanston city, Cook IL
3. Berkeley CCD, Alameda CA
4. Alexandria city, Alexandria city VA
5. Portland West CCD, Multnomah OR
```

Thank you!