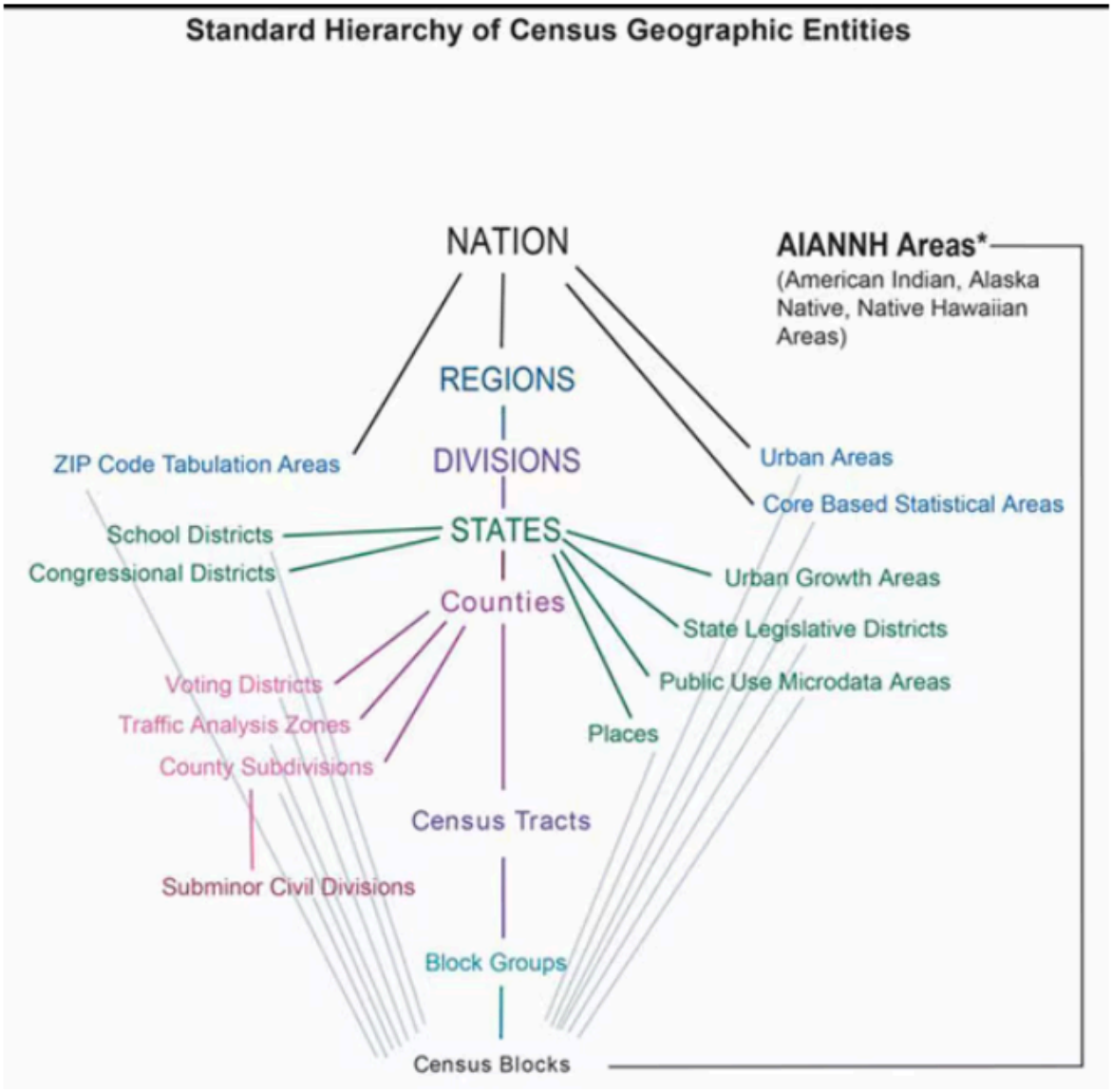
Motivation and Data summary

When I first saw that the data must be over 5000 rows, contain many columns, and target labels, I had a hard time finding one that was at least a little interesting. After some time, I stumbled upon some data from the Environmental Protection Agency (EPA). I had previously worked with some California Environmental data before, which meant that I could use that experience to help analyze this new dataset called the Smart Location Database (SLD). Of all the EPA datasets, I chose SLD because it sort of relates to future decisions that I have to make. For example, when I have to move somewhere for a new job, I would have to consider the characteristics of the county, city, and specific area since there is not that much data when it comes to housing/renting costs. Although the data is quite outdated (2017-2021), I think it serves as a good baseline of what to consider when deciding where to move.

First off, I need to clarify some things about how the U.S. Census defines geographical areas (cities, counties, etc..). Take a look at the following picture:



<https://www.census.gov/newsroom/blogs/random-samplings/2014/07/understanding-geographic-relationships-counties-places-tracts-and-more.html>

For the SLD, each row represents a Census Block Group (CBG), which can be thought of like a basic geographical unit. Going up the hierarchy, we have the Core Based Statistical Area (CBSA), which can be composed of multiple Census Block Groups. Then, we have the Combined Statistical Area (CSA), which can be composed of multiple Core Based Statistical Areas. However, in the analysis, I will mainly be focusing on Census Block Groups even though effects from the CBSA and CSA will bleed through.

Now, because we have to create target labels, I had to decide what kinda questions I wanted to answer. So, I decided to split up California into 4 regions: Bay Area, Northern, Southern, and Central and figure out how well I could classify the Census Block Groups into their respective regions. My intuition is that Census Block Groups in Northern or Central California will be quite different from the rest. I would think that areas in the Bay Area and Southern California would be quite similar. However, it's important to note that not all Census Block Groups lie inside one of the 4 specific regions. I used the following information from a California nonprofit association to assign the Census Block Groups into their regions:

<https://www.calbhbc.org/region-map-and-listing.html>

In terms of the variables used, the EPA provides a large variety to choose from. They range from street density, jobs per worker ratios, jobs per specific industry, traffic amount, and public transit stats. For more information, the Smart Location Database has documentation on their website.

Descriptive Statistics

Alright, before cleaning the data, the dataset has 23212 rows and 182 columns. There are 45 distinct California counties. After attempting to assign each row to a particular California region, the data contained 20873 rows and 184 columns. The 2 extra columns would be the name of the county and name of the region the area is in.

Null Values

Column	# of Null Values
CSA	3505
CSA_Name	3505
CBSA_Name	169
D1C8_OFF	20
Vehicles	2266

Null values for CSA, CSA_Name, and CBSA_Name indicate that some Census Block Groups aren't part of some larger group of Census Block Groups. I don't know if the data is missing because those areas were hard to reach or for some other reason. This would be known as sampling bias. Also, I checked the distribution of the population and land size variables for the rows where CSA, CBSA_Name, and CSA_Name were missing and they were quite "normal" looking. So, the bias for these missing values would not come from how small or big the population/area is. For these reasons, I don't really think that these rows with missing rows will create such a bias that would meaningfully impact my analysis.

Negative Values/Errors

In terms of negative values/errors, a value of -99999 indicated missing data so I removed all rows that contained a single missing value or a value of -999999. Regular negative values like -2.44 or -1 were not considered to be errors following closer inspection. After removing all rows with such values, the data contained 13471 rows and 184 columns.

Distribution of Target Variable Before Removing Nulls/Errors

Region	# of Census Block Groups
Southern California	13193
Bay Area	4178
Central California	2897
Northern California	605

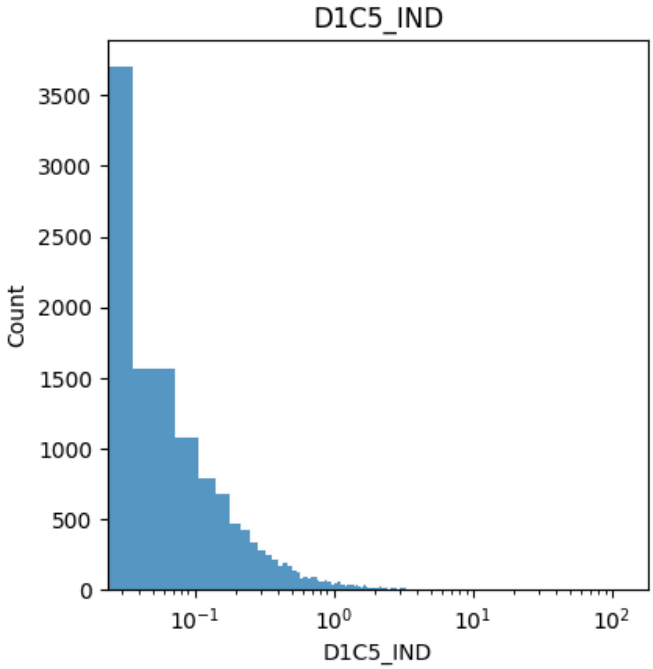
Distribution of Target Variable After Removing Nulls/Errors

Region	# of Census Block Groups
Southern California	8968
Bay Area	2517
Central California	1737
Northern California	249

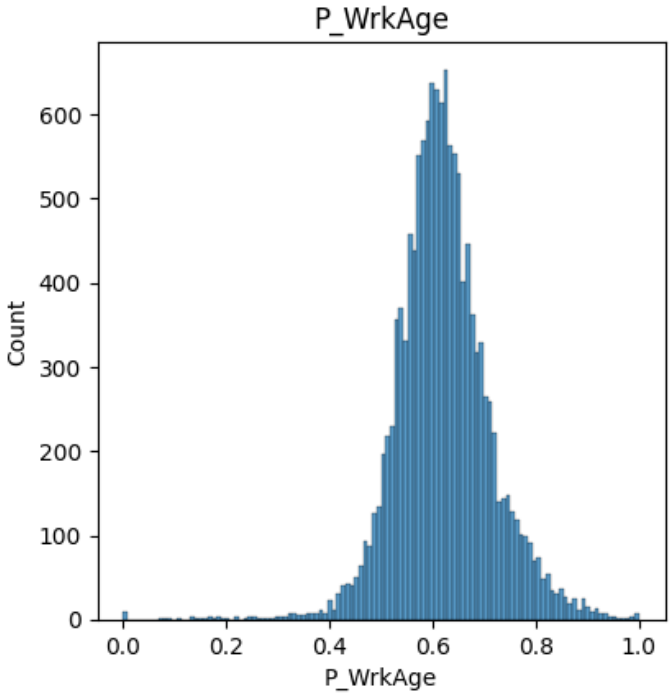
It is no surprise that most of the rows belong to Southern California. We will have to counteract this by using oversampling methods later on.

Distributions

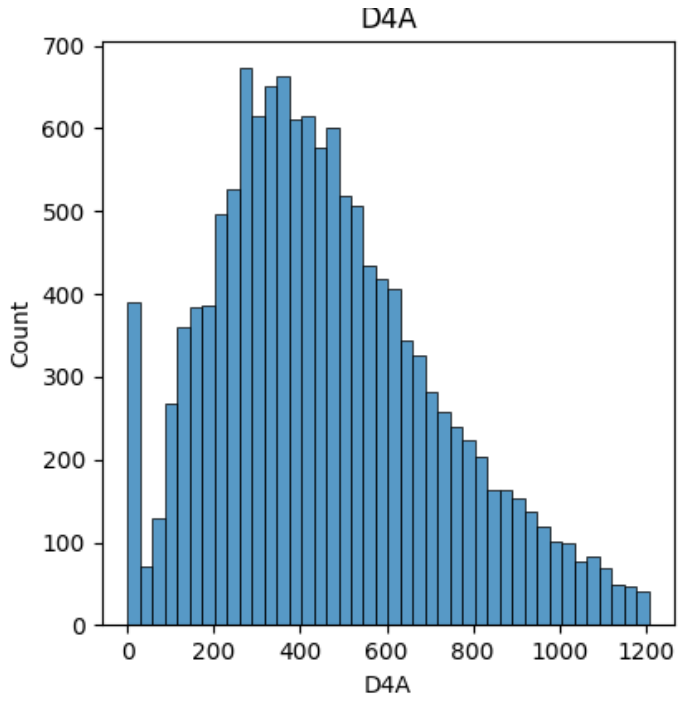
Since there are so many variables, I will choose a few histograms to show for the variables chosen for the actual clustering and supervised learning. All of the following variable definitions are from the Smart Location Database documentation (<https://www.epa.gov/system/files/documents/2023-10/epa_sld_3.0_technicaldocumentationuserguide_may2021_0.pdf>)



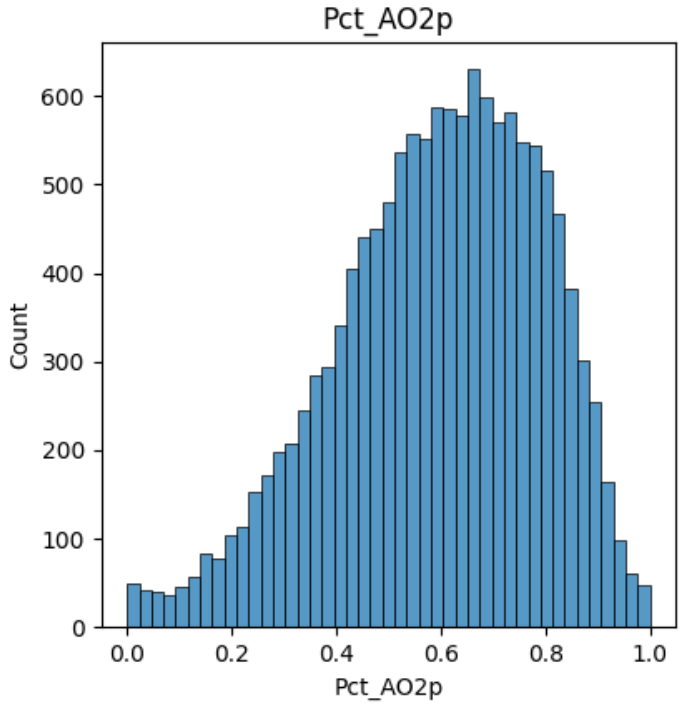
D1c5_Ind	Gross industrial (5-tier) employment density (jobs/acre) on unprotected land



P_WrkAge	Percent of population that is working aged 18 to 64 years,



D4a	Distance from the population-weighted centroid to nearest



Pct_AO2p	Percent of two-plus-car households in CBG, 2018

Various sample statistics

	Ac_Land	Ac_Unpr	P_WrkAge	AutoOwn0	Pct_AO2p	E5_Ret	E5_Ent	E8_Hlth	D1C5_RET	D1C5_IND
count	13471.000000	13471.000000	13471.000000	13471.000000	13471.000000	13471.000000	13471.000000	13471.000000	13471.000000	13471.000000
mean	339.664921	250.658867	0.620634	48.916784	0.593864	82.285502	100.380744	124.242150	0.632095	0.737723
std	4116.565820	1310.140462	0.092262	82.042930	0.197057	252.685108	399.282251	459.071392	1.961353	2.917049
min	3.806823	3.806823	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	64.009278	62.491096	0.567000	8.000000	0.464562	0.000000	0.000000	17.000000	0.000000	0.029345
50%	106.410403	102.778227	0.616000	25.000000	0.612903	13.000000	18.000000	37.000000	0.111695	0.123894
75%	183.090845	174.740024	0.670000	60.000000	0.744975	63.000000	84.000000	89.000000	0.538239	0.455436
max	407868.018652	61946.753520	1.000000	1710.000000	1.000000	5780.000000	27592.000000	12572.000000	84.893696	121.727268

Final Model Description and Result

Before doing feature selection, I made sure to delete any variables that gave away too much information. What I mean by this is that the variable only has a few distinct values. What this means is that many of the Census Block Groups share the **same value**. This is not what we want since we want to predict regions based on the Census Block Group. However, areas that are close to each other will likely have values close to one another. We want to select variables that contain as little as information from **up above the hierarchy**. Remember, it goes like Census Block Groups → Core Based Statistical Area → Combined Statistical Area. The reason we want to use this kind of approach is because we would basically be cheating since one row represents one Census Block Group.

Feature Selection

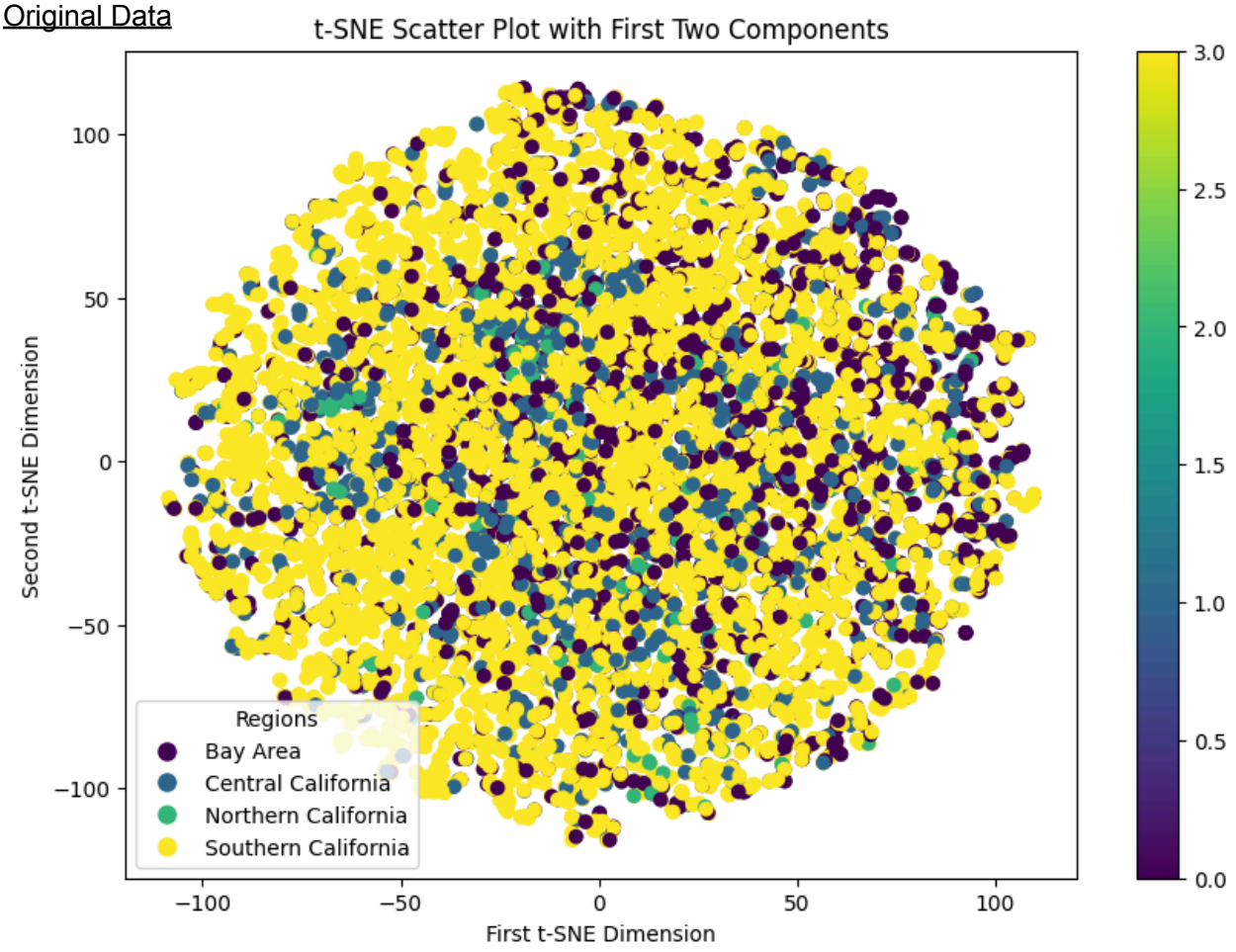
As for the actual feature selection method, I used a linear correlation approach (pearson's correlation) since I am dealing with numerical variables and I have some intuition that many of the variables will carry information above the level of detail that we want (CBG). I used a correlation threshold approach to deal with both issues. However, I modified it to be an

iterative approach in which I find all the correlation pairs above the threshold and randomly remove one of the variables. Then, I produce the correlation matrix again after deleting the variables randomly chosen and apply the threshold approach. The loop continues until all correlation pairs don't have a value > threshold. I used a threshold of 0.7 after trying a threshold of 0.5, 0.6, and 0.9. I found 0.7 to provide the best results when it came to clustering and multi-class classification later on. That would leave me with 32 columns to work with.

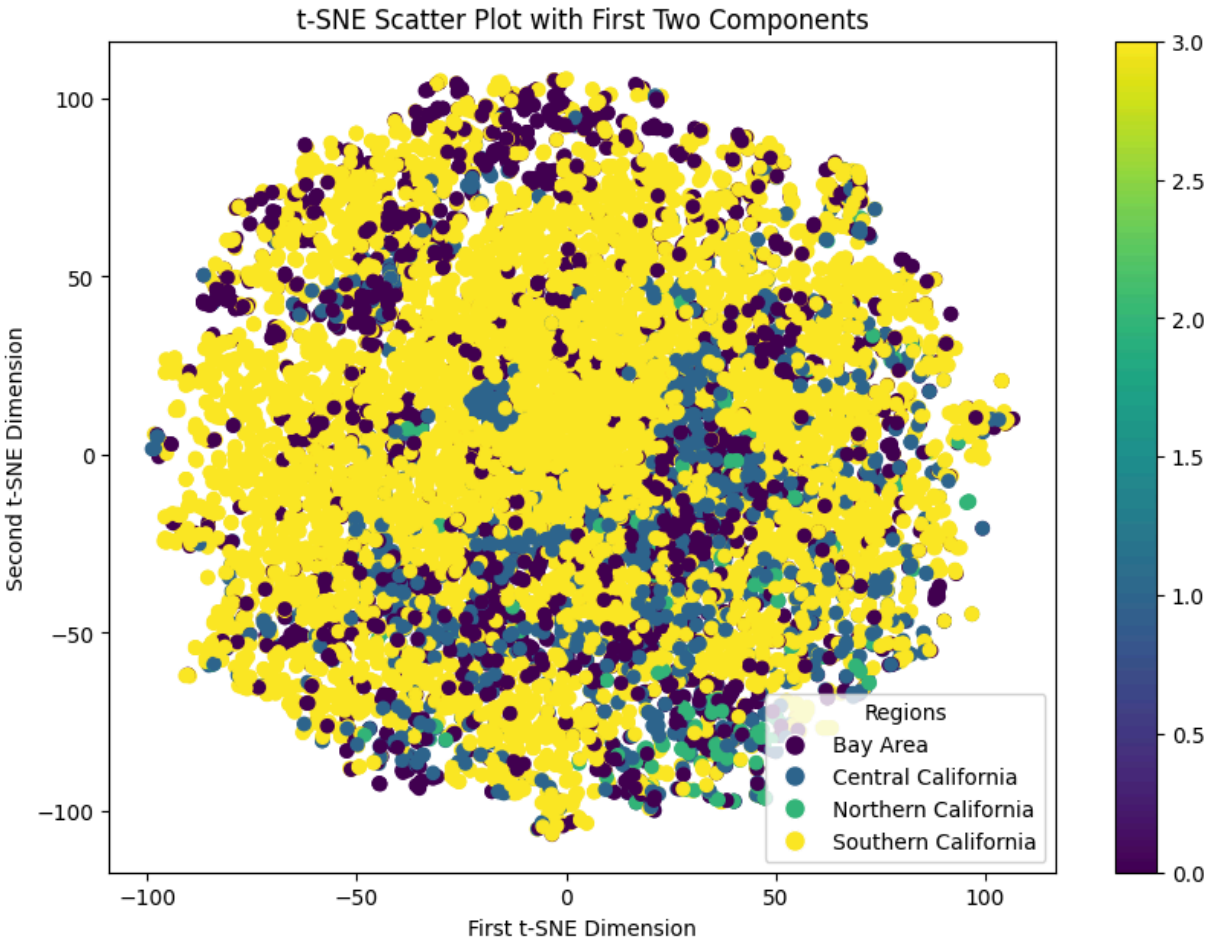
Dimensionality Reduction

Before applying any oversampling/undersampling, I wanted to see how PCA t-SNE would perform. MDS, LLE, and IsoMap were not used because the computing time was far too long in my opinion. Here are the results for how t-SNE performed with a low, medium, or high correlation threshold (0.5, 0.7, 0.9).

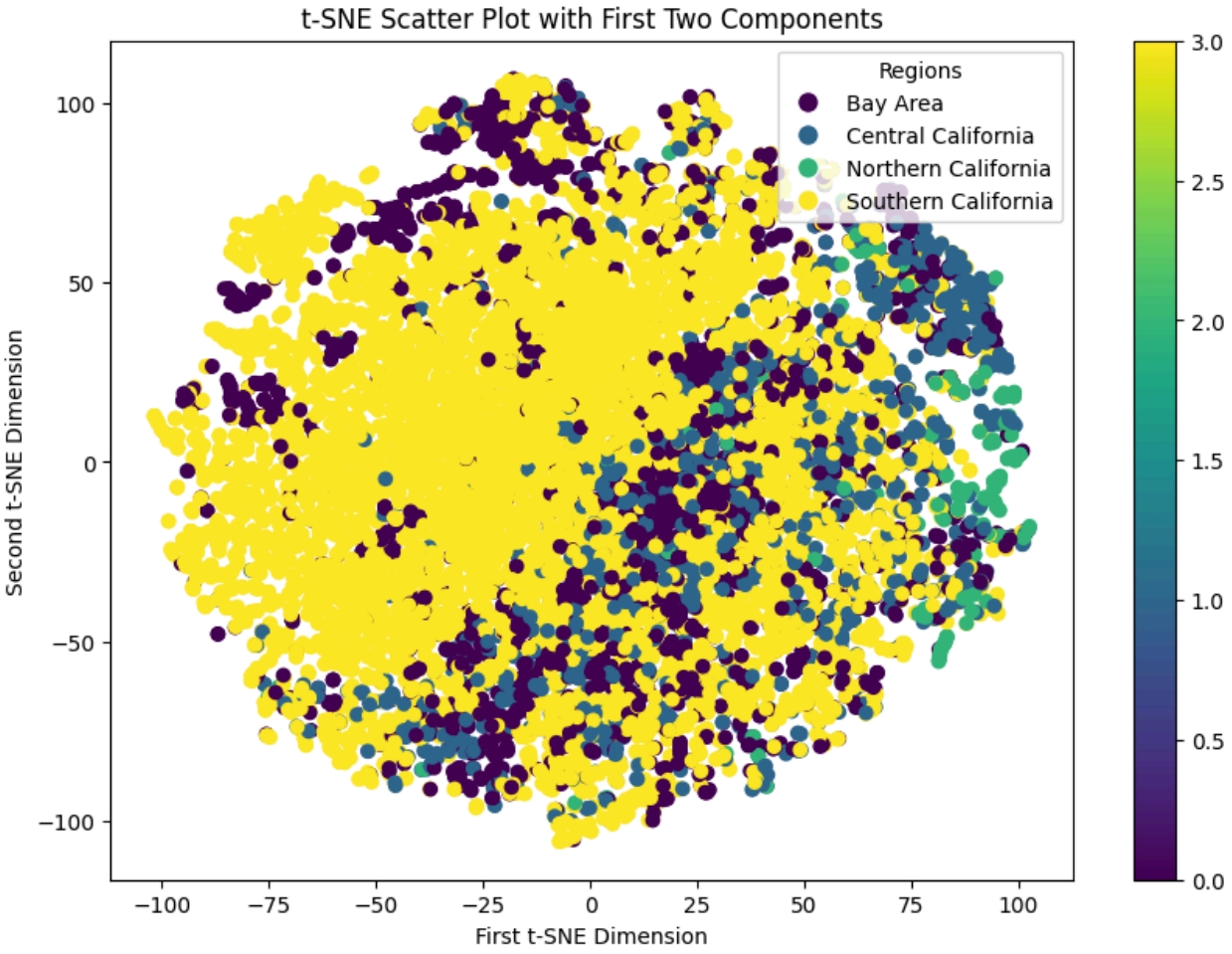
Original Data



Correlation Threshold = 0.5



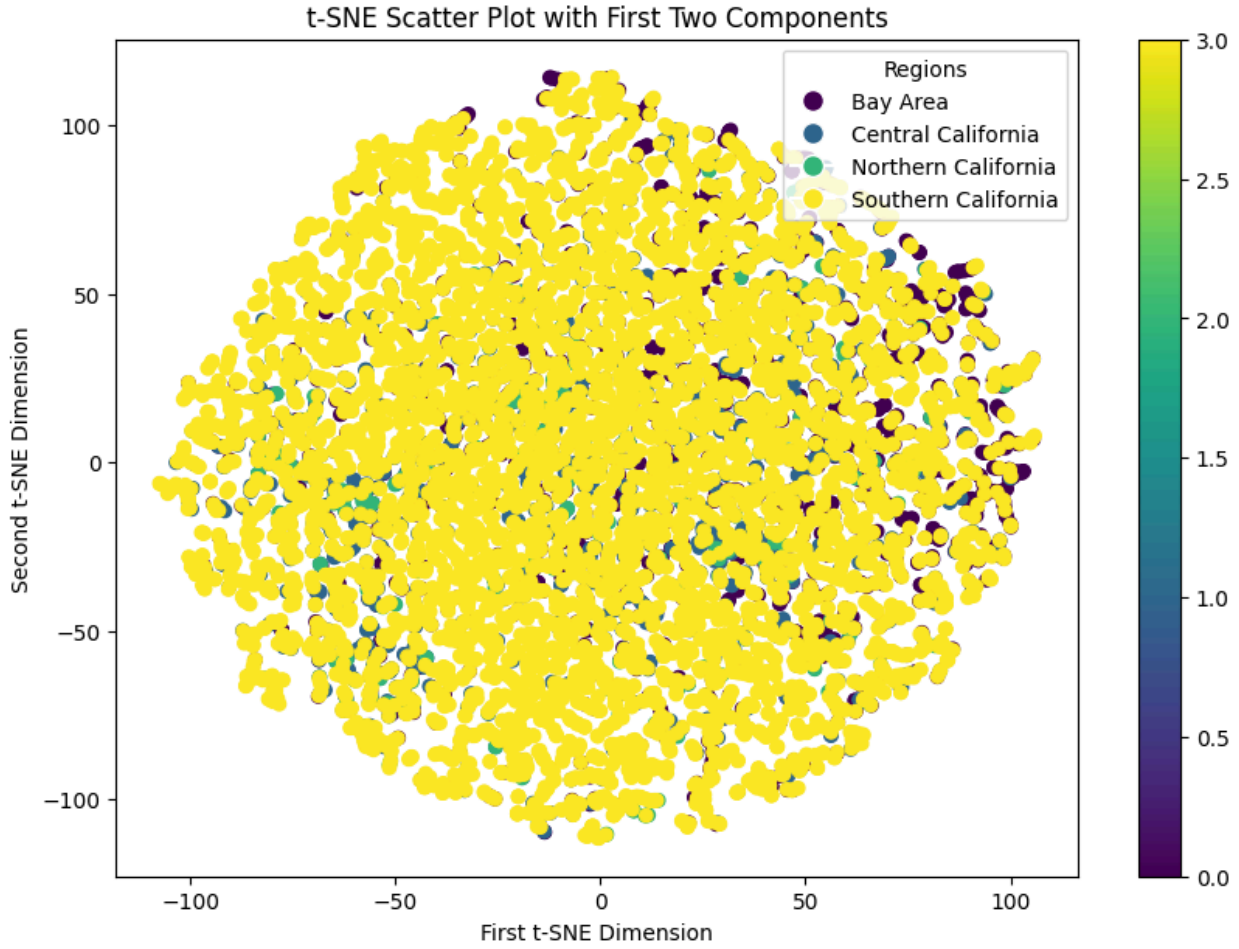
Correlation Threshold = 0.7



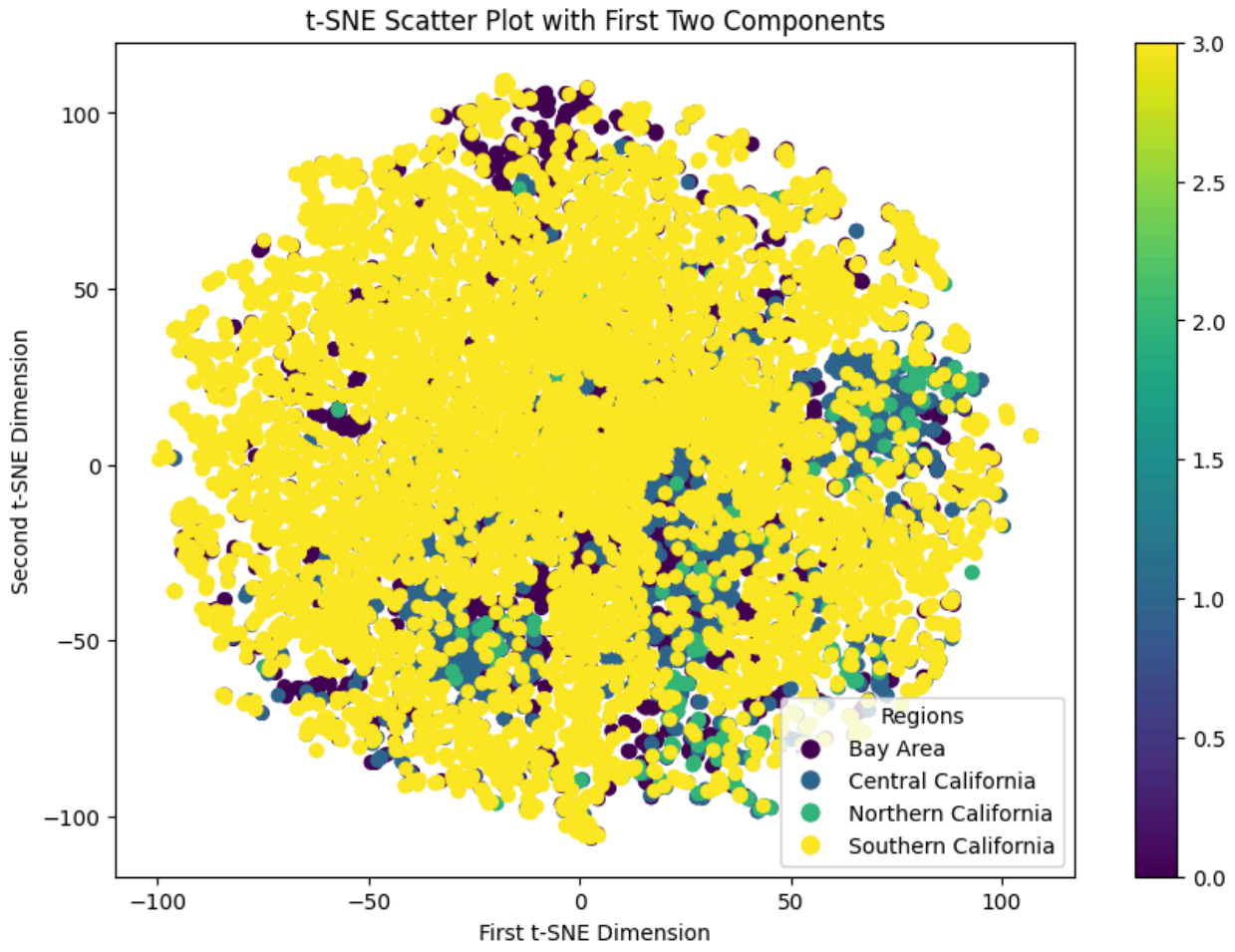
Correlation Threshold = 0.9

Undersampling

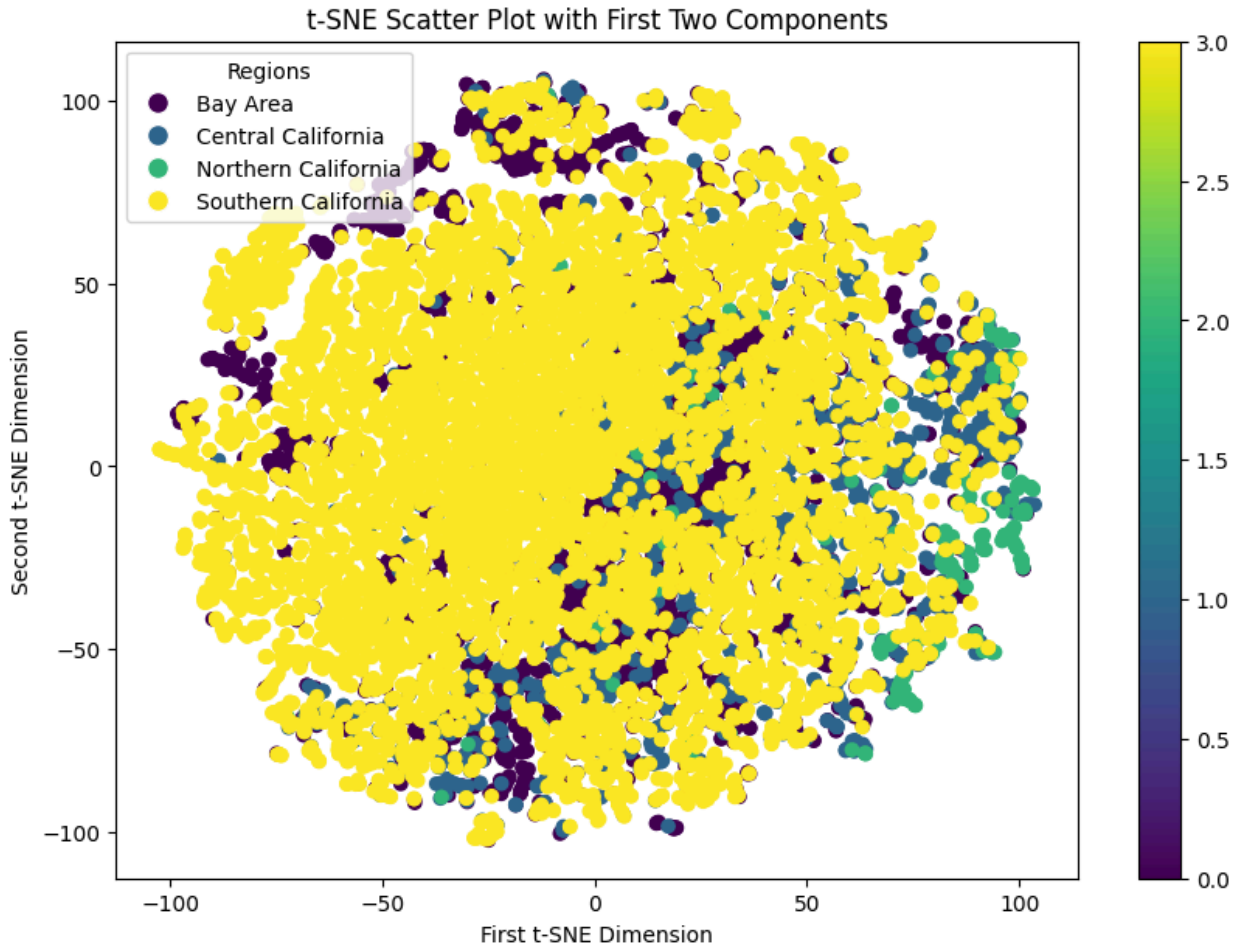
Default OneSidedSelection was used; it uses the ‘not-minority’ argument to decide what classes to undersample from. I also used the ‘majority’ argument to only undersample Southern California but results were pretty similar in terms of horrible performance.



Correlation Threshold = 0.5



Correlation Threshold = 0.7

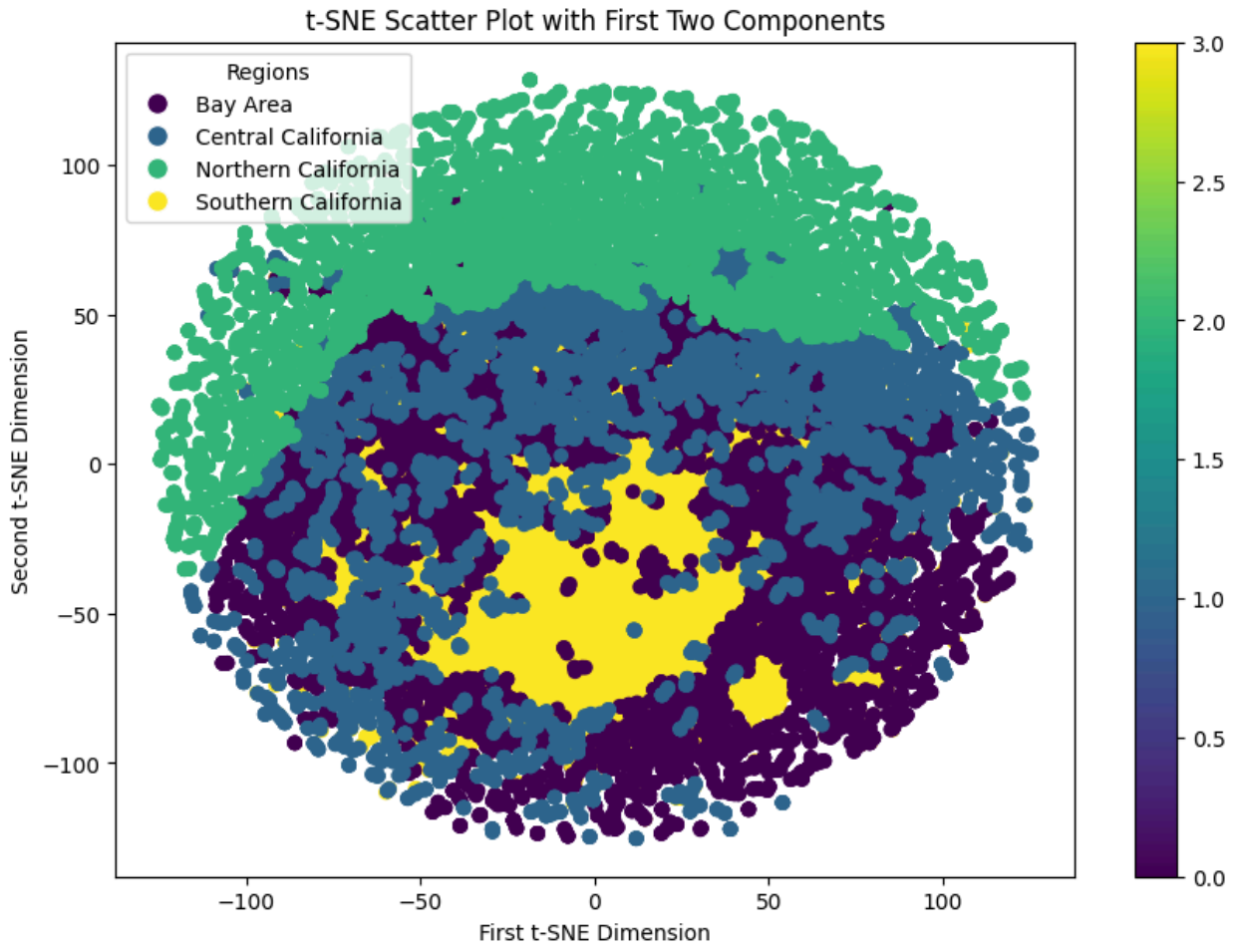


Correlation Threshold = 0.9

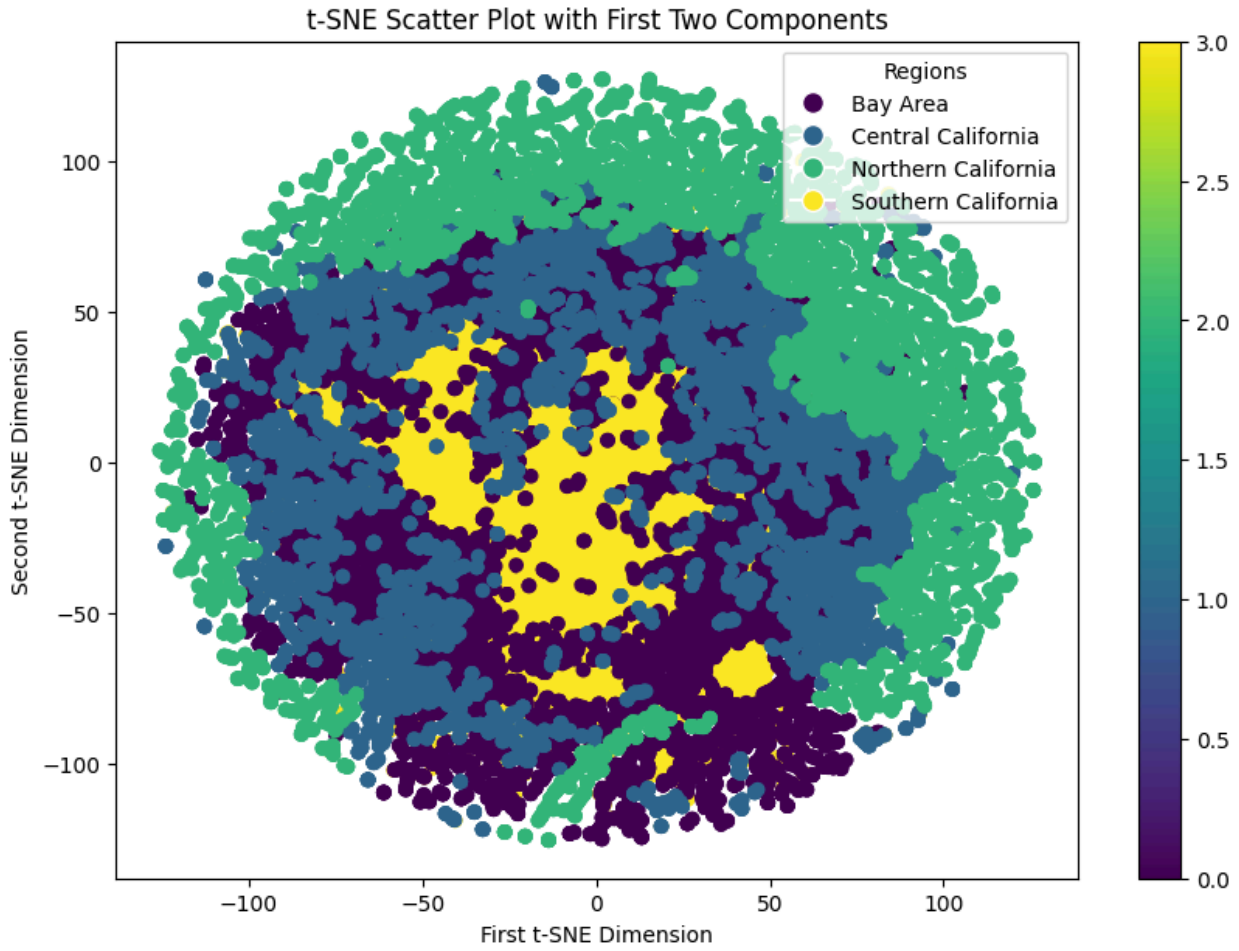
Oversampling

Oversampling gave the most hopeful performance in terms of dimensionality reduction. At first, I used regular Smote and it gave decent results. Then, I used the BorderlineSmote with a correlation threshold of 0.7 after manually running some combinations and found the following combination to be quite solid: BorderlineSMOTE(sampling_strategy='not majority', m_neighbors=200, k_neighbors=200).

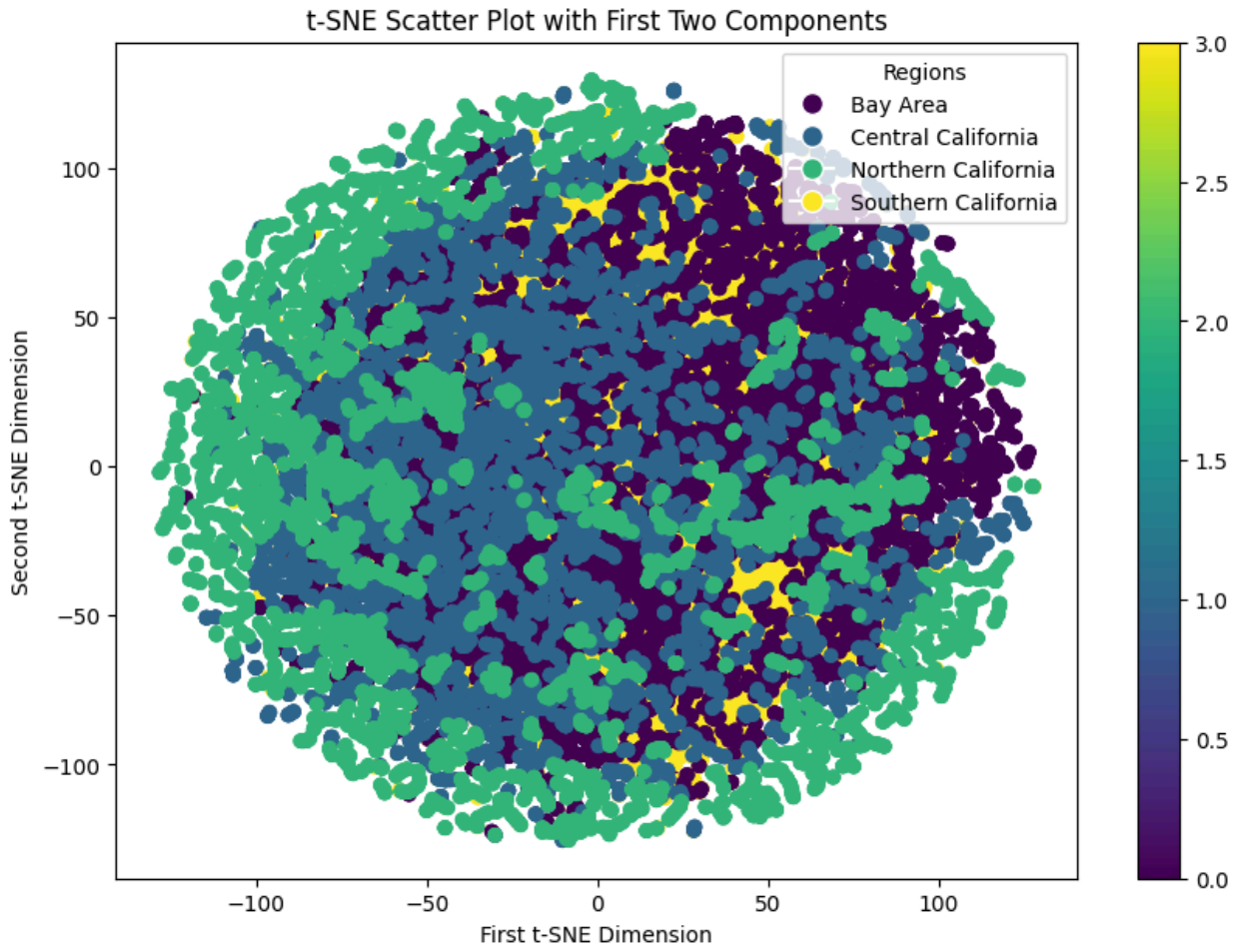
Regular Smote



Correlation Threshold = 0.9

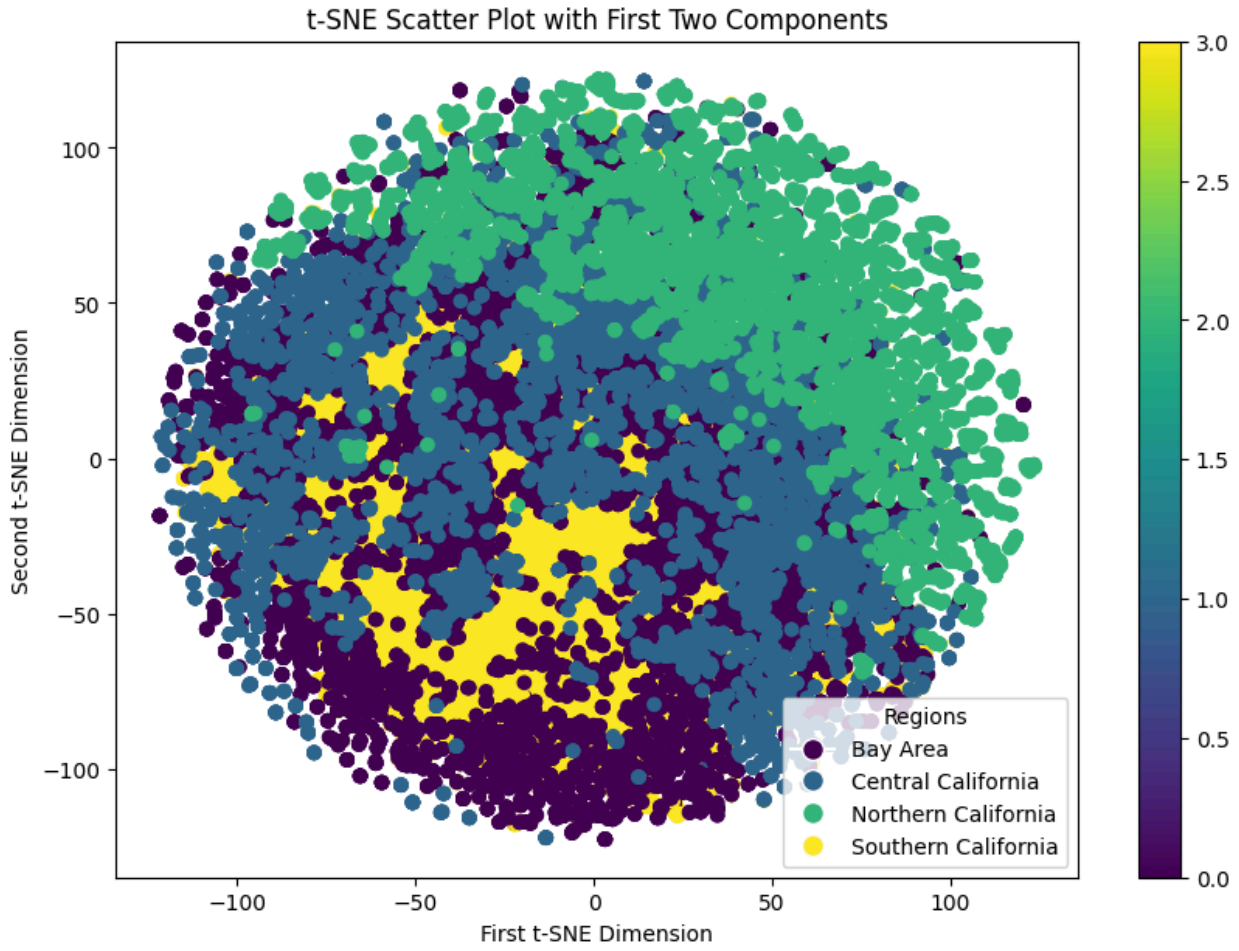


Correlation Threshold = 0.7

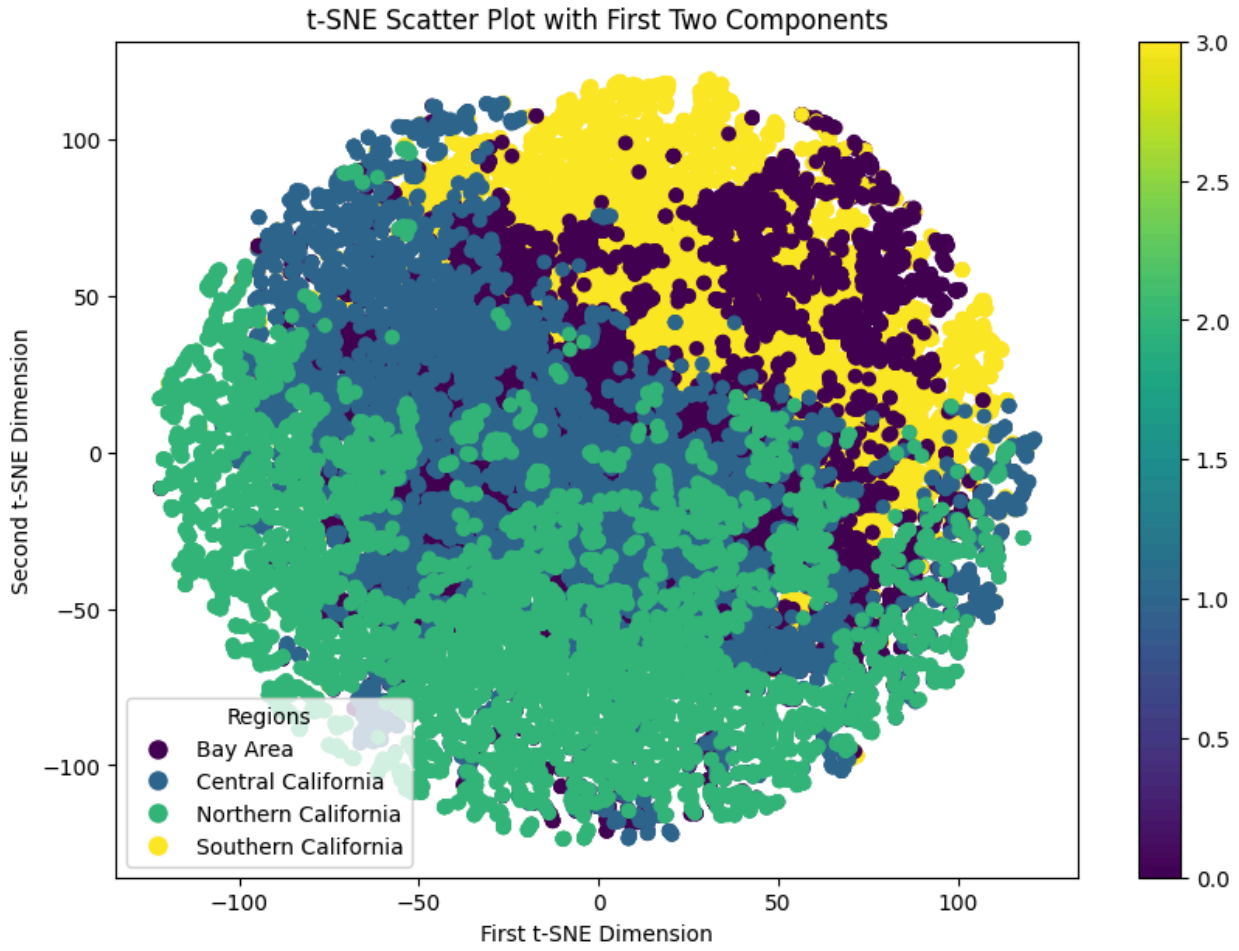


Correlation Threshold = 0.5

BorderlineSmote



Correlation Threshold = 0.7



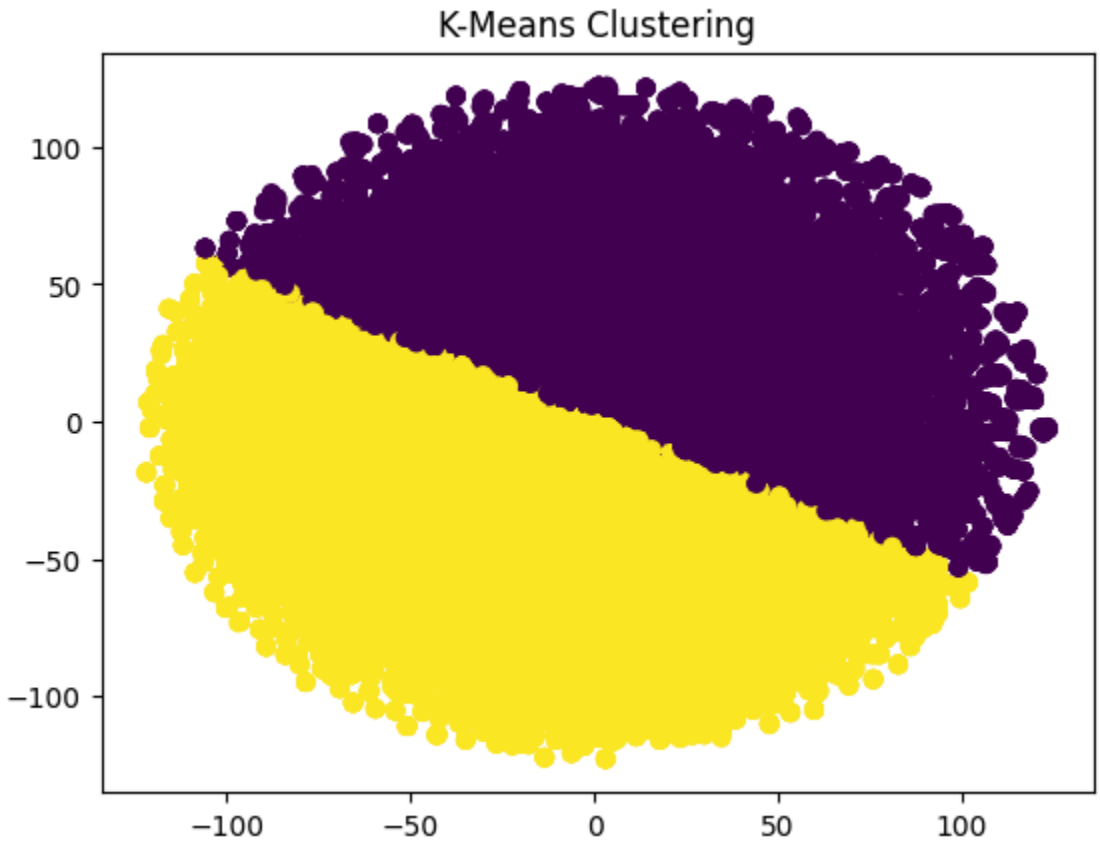
Correlation Threshold = 0.7 and Top 5 Features after Random Forest Feature Importance

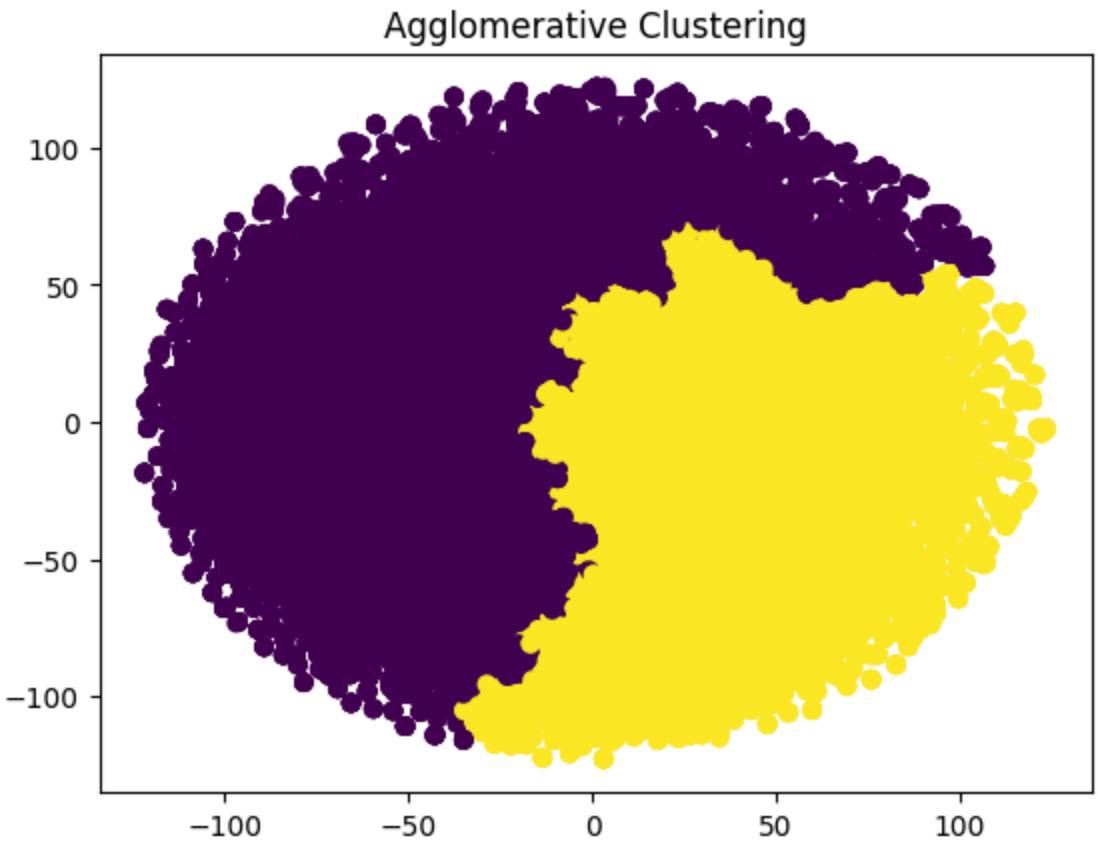
Clustering

Clustering methods left a lot to be desired but considering the complexity of the data and how the data looked after dimensionality reduction, this should come to no surprise. I do not show results when clustering on the original feature space due to the variables giving too much information away when it comes to predicting the region. There were many redundant and highly correlated variables which led to a complete disaster when it came to clustering.

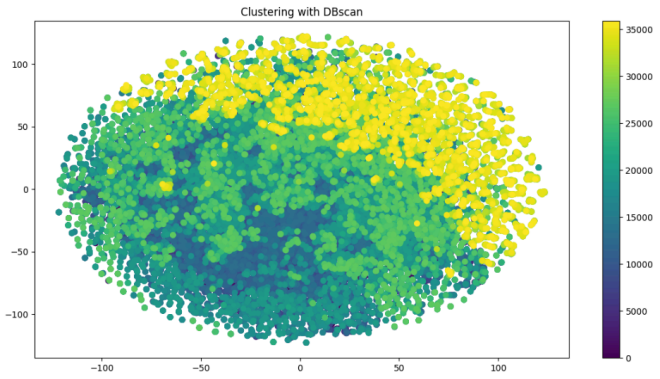
Default Clustering on reduced feature space

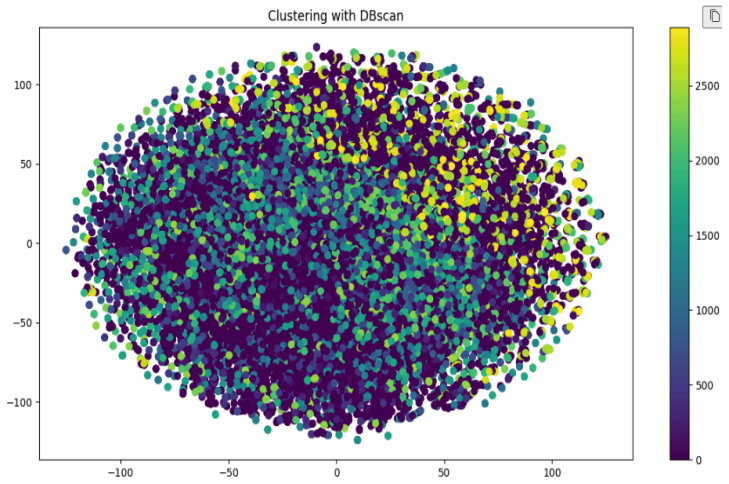
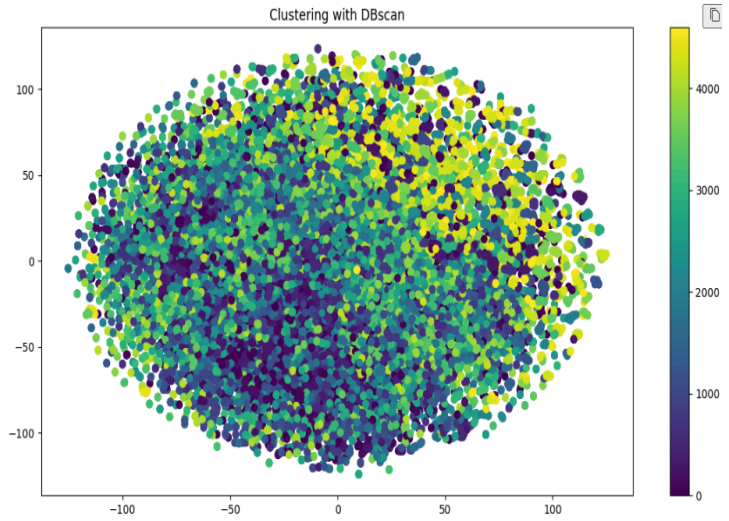
I tried default K-means with two clusters and agglomerative clustering (Linkage = Ward) with two clusters also. I did not expect much from these two to be honest.

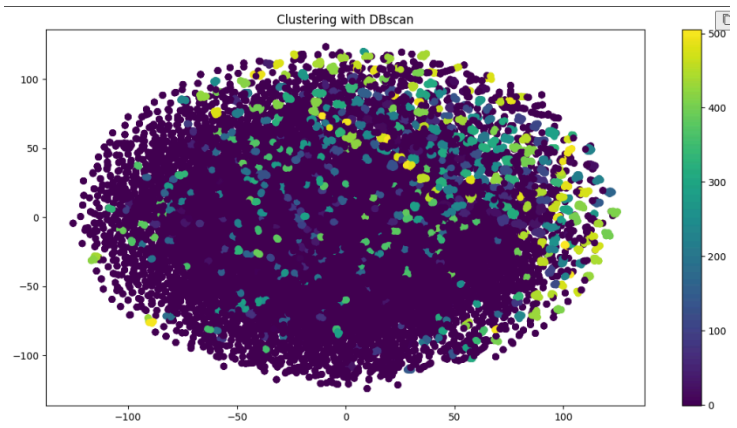




Suggested Clustering

Clustering Configuration	Scatter Plot
DBSCAN(eps=0.00001,metric='euclidean', min_samples=1)	

DBSCAN(eps=0.5,metric='euclidean', min_samples=5)	
DBSCAN(eps=0.9,metric='euclidean', min_samples=3)	

DBSCAN(eps=1.2,metric='euclidean', min_samples=12)	

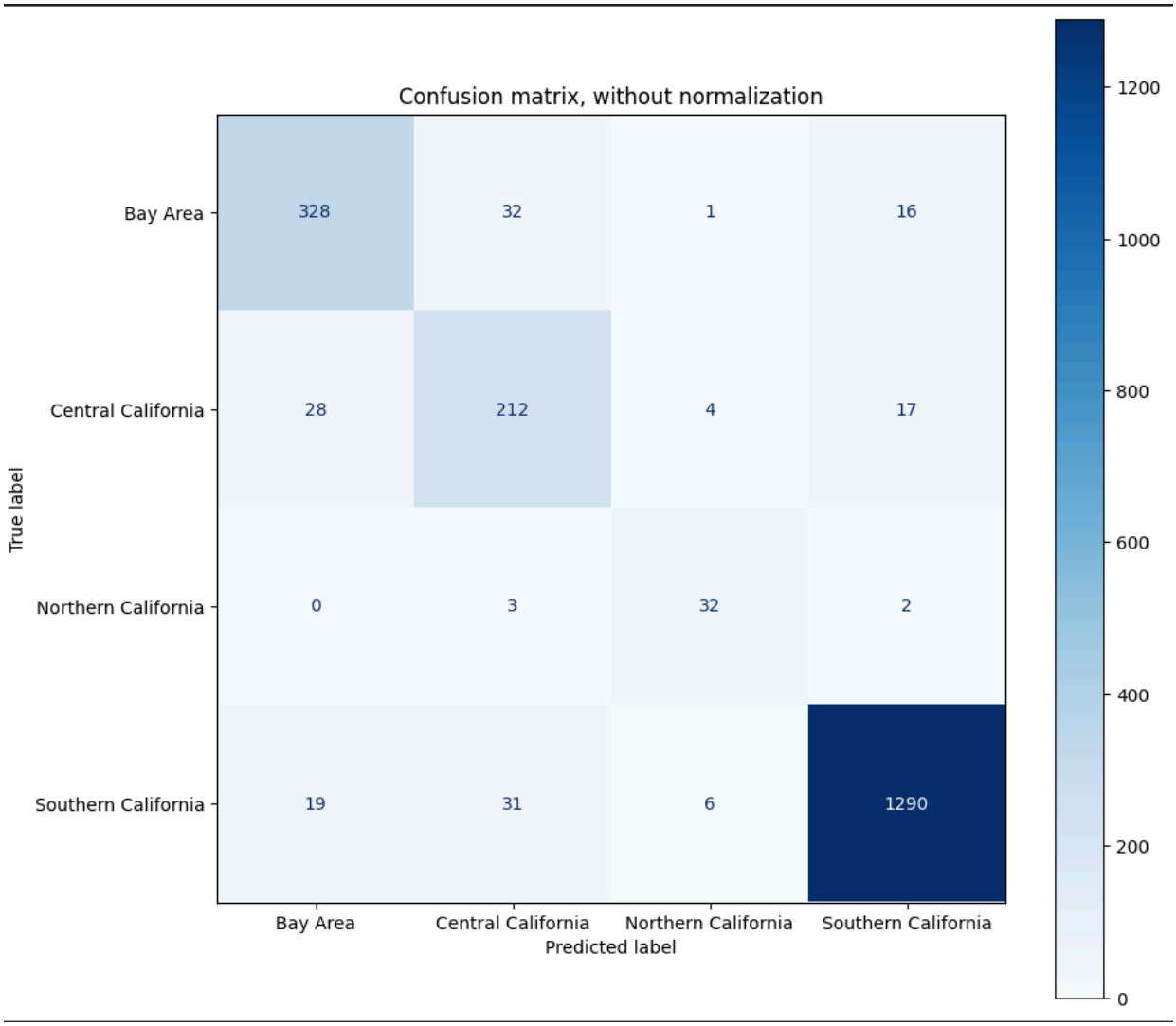
Using any other configuration led to complete disaster. What happened is that as I increased the eps and min_samples parameters, one of the groups would completely dominate/overlap on top of all the other points. Changing distance metric also did not change much. The first configuration seems to be the best option even though DBSCAN doesn't "directly" say 2,3, or 4 clusters. Because we did so much preprocessing and transformations, I don't think I could quantitatively talk about the patterns from the clustering/plots. I can only talk about things like shape, structure, or density.

Discussion (analysis)

For the target labels, I did multi-class classification with Random Forest and Xgboost (multiclass). I used StandardScaler and the following oversampling configuration: BorderlineSMOTE(sampling_strategy='not majority', m_neighbors=100, k_neighbors=100). Here are the k-fold cross validation results:

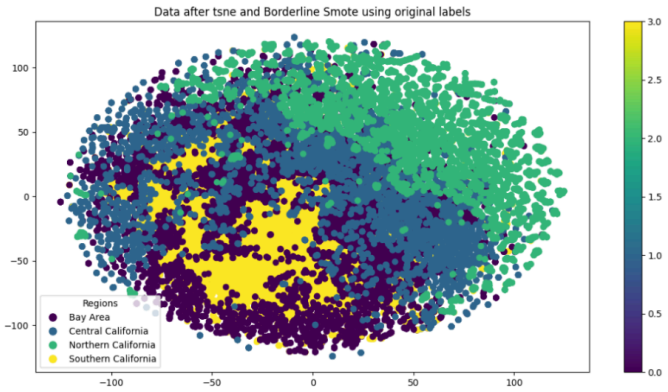
Models	Mean_score (f1_weighted)	Std_score (f1_weighted)
Random Forest	0.85701	0.006256
XgBoost(multi class)	0.905314	0.006882

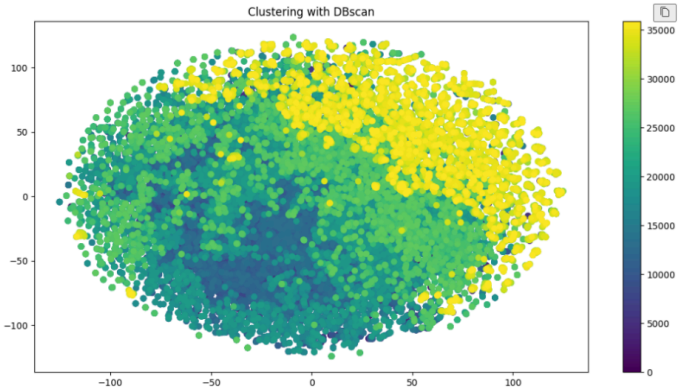
Using a separate validation set:



	precision	recall	f1-score	support
0	0.8747	0.8700	0.8723	377
1	0.7626	0.8123	0.7866	261
2	0.7442	0.8649	0.8000	37
3	0.9736	0.9584	0.9659	1346
accuracy			0.9213	2021
macro avg	0.8388	0.8764	0.8562	2021
weighted avg	0.9237	0.9213	0.9223	2021

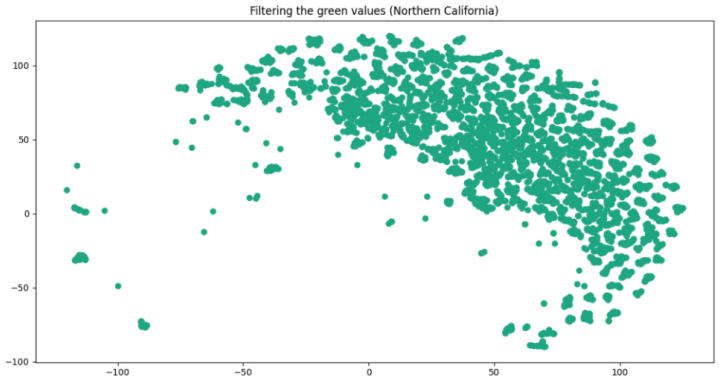
Clustering vs. Original Labels (Reduced Feature Space)

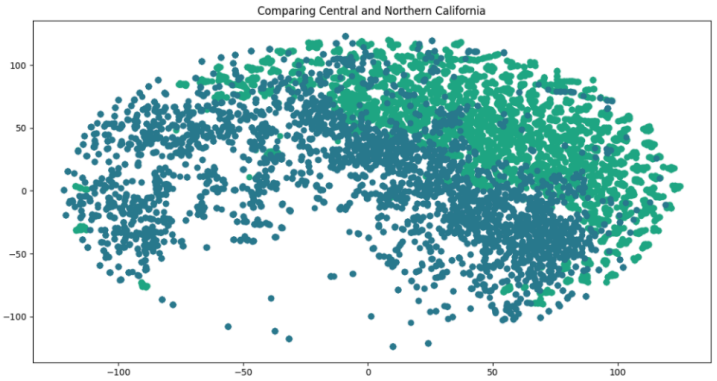


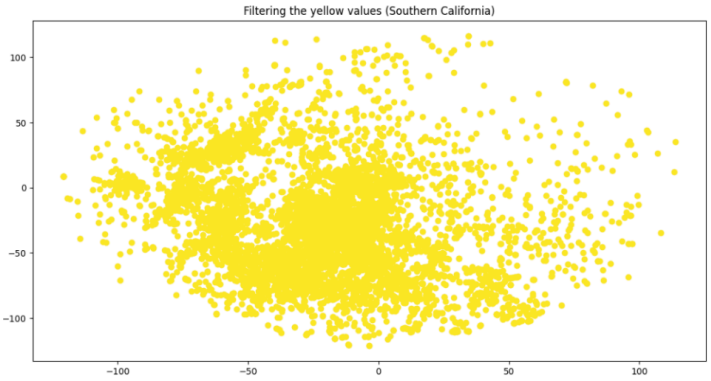


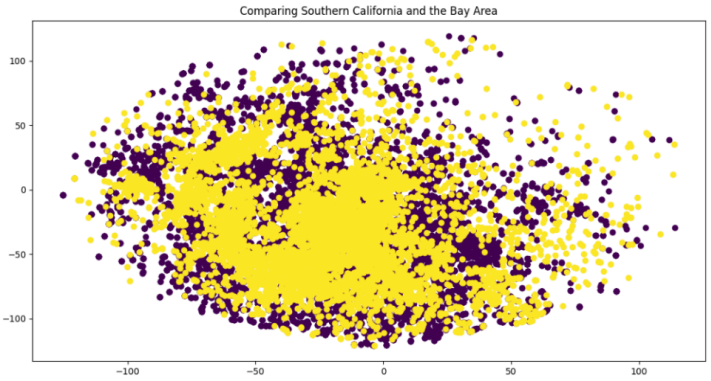
In my opinion, clustering data did not provide anything that the original label scatter plot didn't. I chose t-SNE for its ability to break down complex nonlinear high-dimensional data and used DBscan since it was the only clustering method that even came close to giving something useful.

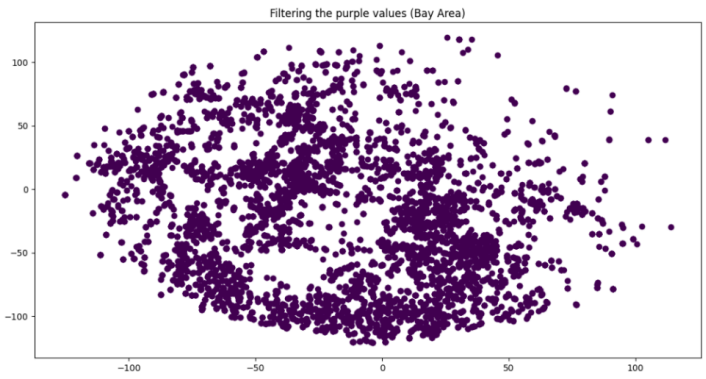
Results in a Domain

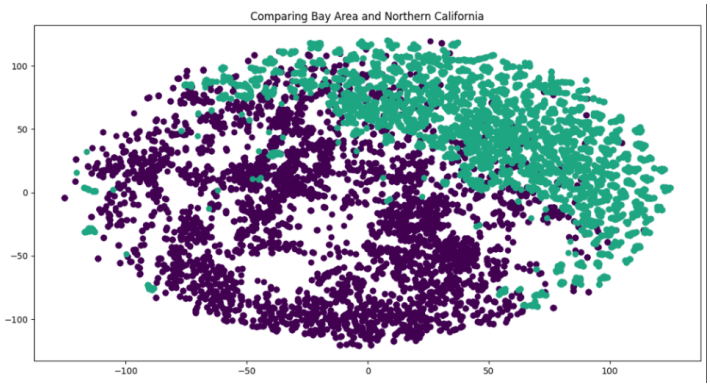


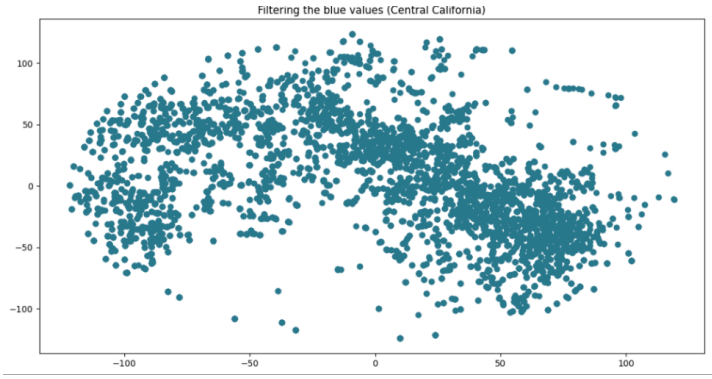


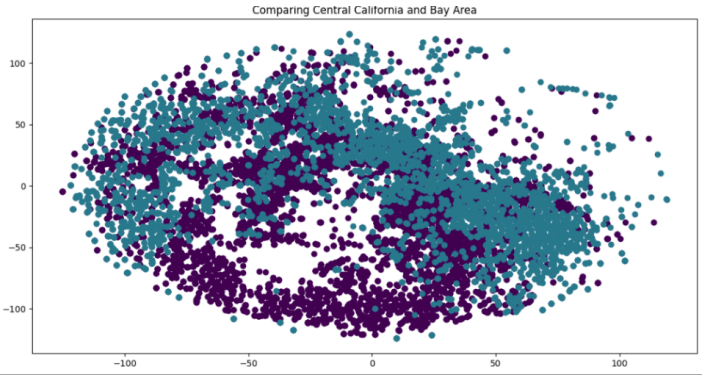












I know the pictures are hard to see but the general conclusion is that Census Block Groups in Northern California are quite different from the rest of the regions. The Bay Area, Central California, and Southern California seem to be similar in structure/shape but differ in density.

In addition, I also created a sort of ranking system where I created a job + traffic (foot and car traffic) category where each category contains some variables about jobs or traffic. I

used MinMaxNormalization so the scores will be bounded by [0,1]. Here are some of the results for counties in Southern California. It might seem counterintuitive that Los Angeles County would have a high score for traffic but the traffic score also takes into account how “walkable” and accessible a county is in terms of traveling. Things like public transport and walking to and from places. Finally, the main finding I discovered while doing clustering and creating a pseudo-ranking system is that census block groups are quite homogenous. There is not a “anomaly” when it comes to a Census Block Group that is doing way better than others. Of course, the data is skewed by Census Block Groups that are a hotspot when it comes to businesses. This can be seen with Los Angeles County dominating the rankings. San Bernardino and Orange County seem to be in the middle of the pack.

COUNTY_NAME	
Los Angeles County	0.002930
Santa Barbara County	0.001034
Orange County	0.000894
San Diego County	0.000680
San Bernardino County	0.000369
Riverside County	0.000317
San Luis Obispo County	0.000307
Kern County	0.000265
Imperial County	0.000107

Job Score

COUNTY_NAME	
Los Angeles County	0.642654
Orange County	0.584887
San Luis Obispo County	0.557148
San Bernardino County	0.552887
Riverside County	0.551792
Santa Barbara County	0.549674
Kern County	0.543568
San Diego County	0.542136
Imperial County	0.531919

Traffic Score

COUNTY_NAME	
Los Angeles County	0.322792
Orange County	0.292890
San Luis Obispo County	0.278727
San Bernardino County	0.276628
Riverside County	0.276054
Santa Barbara County	0.275354
Kern County	0.271917
San Diego County	0.271408
Imperial County	0.266013

Job + Traffic Score