Aaron Miller

This project aims to find which groups of the population are more susceptible to diabetes and pre-diabetes.

I use clustering methods to find naturally occurring groups within the population based on various demographic and health metrics. The dataset I used was derived by Alex Teboul from the results of the Center for Disease Control's Behavioral Risk Factor Surveillance System phone survey from the year 2015. The original CDC datasets have over 400 features and detail a wide range of health metrics. The Alex Teboul version retains only 22 features with the specific purpose of focusing primarily on the relationship of certain specific features with the occurrence of diabetes or pre-diabetes. The feature groups are as follows:

- CLASSIFICATION FEATURE
  - 'Diabetes_012' (0 = No Diabetes, 1 = Pre-Diabetes, 2 = Diabetes)
- HEALTH FEATURES
  - High Blood Pressure? (T/F)
  - High Cholesterol? (T/F)
  - Cholesterol check within past five years? (T/F)
  - Body Mass Index
  - Somked at least 100 cigarettes in life? (T/F)
  - Ever had a stroke? (T/F)
  - Ever had coronary heart disease or myocardial infarction? (T/F)
  - Physical activity in past 30 days? (T/F)
  - Consume fruits at least one time per day? (T/F)
  - Consume vegetables at least one time per day? (T/F)
  - Heavy drinker? (T/F)
  - Possess any sort of healthcare coverage? (T/F)
  - Was there a time in the past 12 months when you needed to see a doctor but could not because of cost? (T/F)
  - General Health (scale of 1-5, 1 = Excellent, 5 = Poor)
  - Number of days in past 30 days was mental health not good
  - Number of days in past 30 days was physical health not good
  - Difficulty walking or climbing stairs? (T/F)
- DEMOGRAPHIC FEATURES
  - Age (in 5-year bins from ages 18 to ≥60)
  - Sex (M/F)
  - Education (scale of 1-8)
  - Income (scale of 1-8, 1 = ≤$10k, 8 = ≥$75k)

EDA and Data Cleaning

The first step in exploring the data is to see whether the classes are balanced in the classification feature 'Diabetes_012'. The table to the right indicates the classes are not balanced. 84% of the sample population are in class 0 (i.e. no diabetes), while only 2% are in class 1 and 14% are in class 2. This indicates we will need to use over-sampling and/or under-sampling methods to balance the classes.

	Diabetes_012
0.0	0.84
1.0	0.02
2.0	0.14

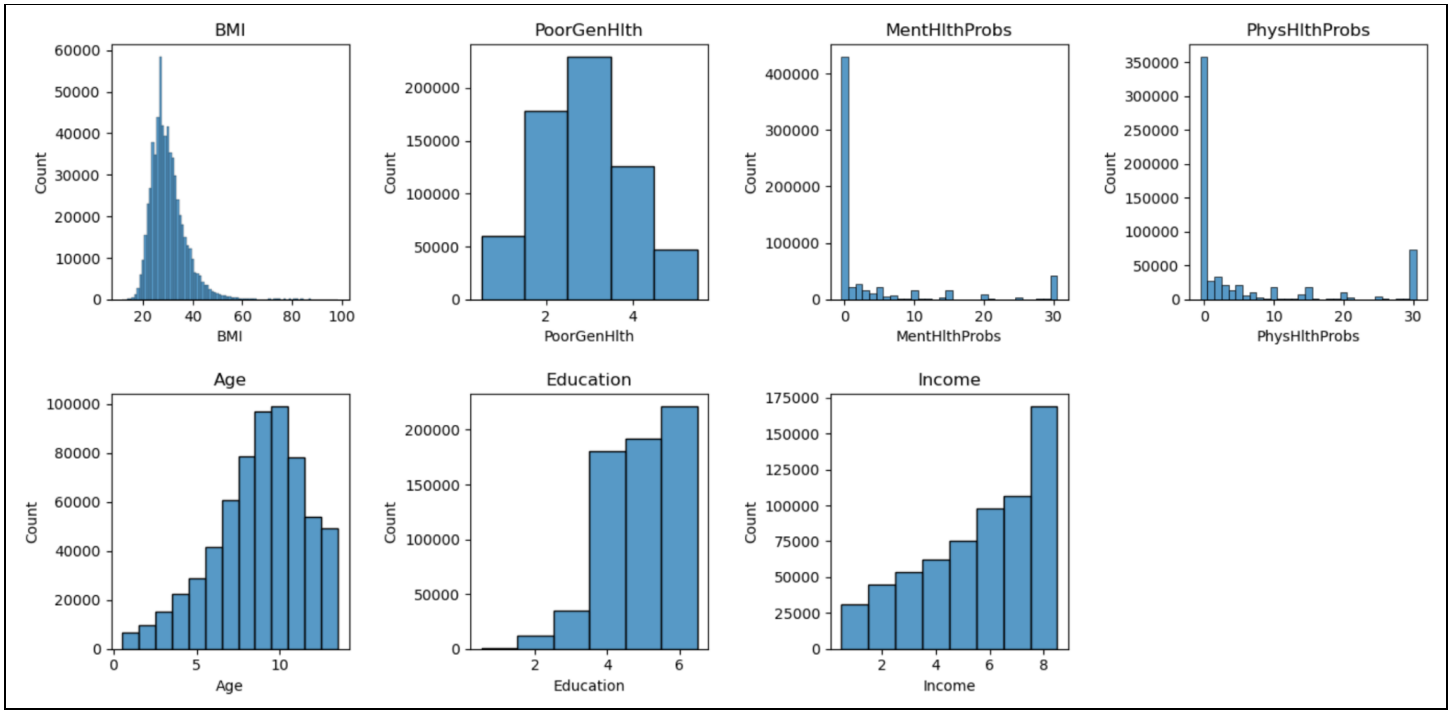
In order to balance the classes, I chose to oversample the minority classes because there are few records in the minority class. Therefore, it makes more sense to increase the minority class than to reduce the majority class to a small number of records. We see to the right the count of records for each class before and after over-sampling was performed. This shows the over-sampling was successful and we have an equal amount of records for each class.

```
0.0    213703
2.0     35346
1.0      4631
Name: Diabetes_012
```

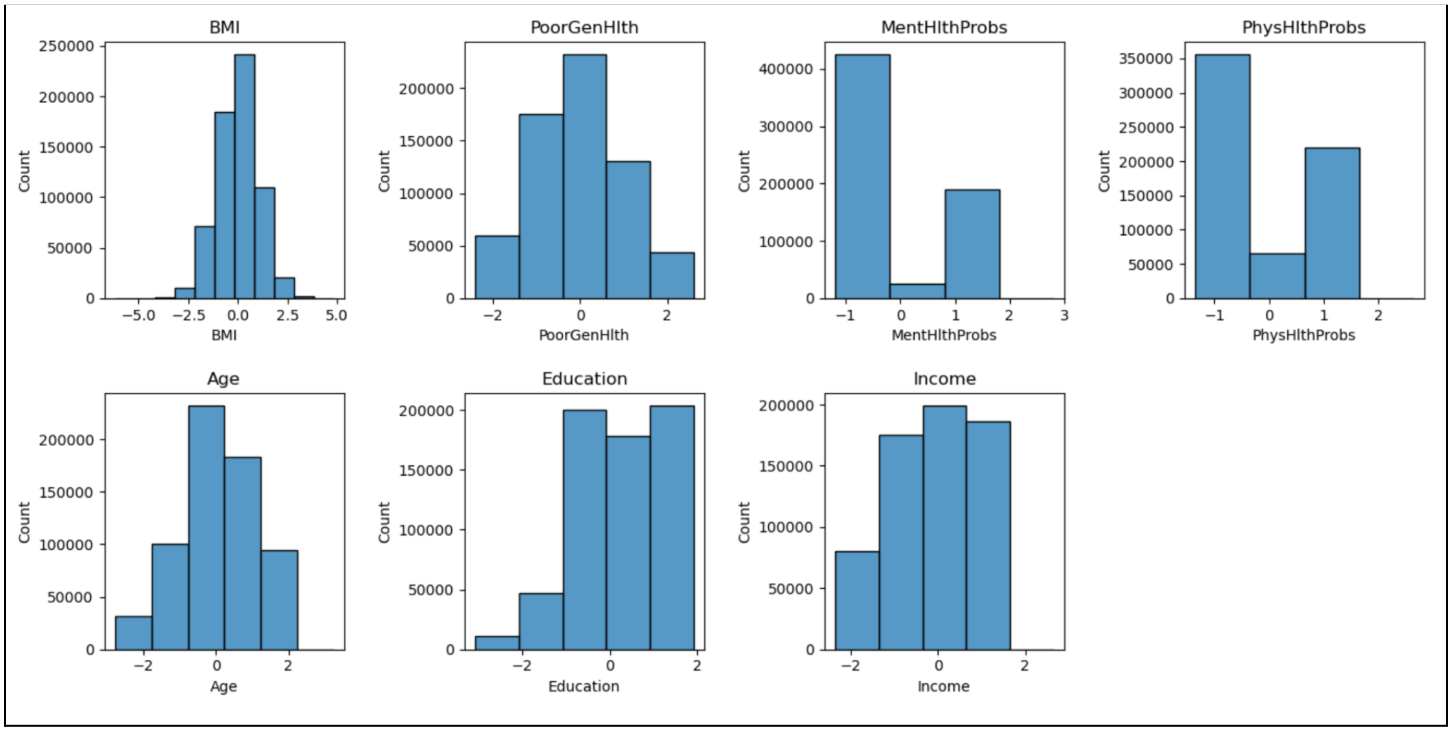
```
0.0    213703
2.0    213703
1.0    213703
Name: Diabetes_012,
```

Looking at general statistics of the features and at the histograms of the continuous/scalar features (both shown below), we see many instances of imbalanced classes and skewed data. For example, ‘Stroke’ has over 75% of records in class 0, while ‘PhysActivity’ has under 25% of records in class 0. For the continuous/scalar features, ‘BMI’, ‘MentHlthProbs’ and ‘PhysHlthProbs’ have a strong right skew while ‘Education’ and ‘Income’ have a strong left skew.

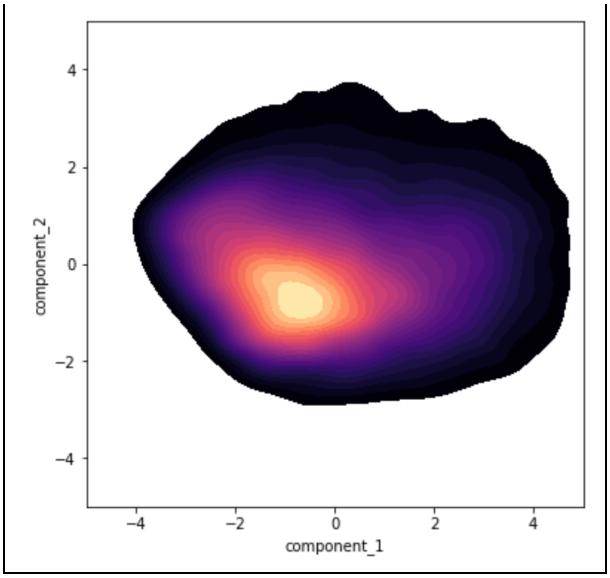
	count	mean	std	min	25%	50%	75%	max
HighBP	641109.0	5.928914e-01	0.463061	0.000000	0.000000	1.000000	1.000000	1.000000
HighChol	641109.0	5.662793e-01	0.464096	0.000000	0.000000	0.865196	1.000000	1.000000
CholCheck	641109.0	9.804856e-01	0.132628	0.000000	1.000000	1.000000	1.000000	1.000000
Smoker	641109.0	4.763032e-01	0.465408	0.000000	0.000000	0.380585	1.000000	1.000000
Stroke	641109.0	5.317453e-02	0.201154	0.000000	0.000000	0.000000	0.000000	1.000000
HeartDiseaseorAttack	641109.0	1.370049e-01	0.314503	0.000000	0.000000	0.000000	0.000000	1.000000
PhysActivity	641109.0	7.068866e-01	0.421442	0.000000	0.257564	1.000000	1.000000	1.000000
Fruits	641109.0	6.186637e-01	0.451781	0.000000	0.000000	1.000000	1.000000	1.000000
Veggies	641109.0	7.937970e-01	0.372371	0.000000	0.791369	1.000000	1.000000	1.000000
HvyAlcoholConsump	641109.0	3.916107e-02	0.180638	0.000000	0.000000	0.000000	0.000000	1.000000
AnyHealthcare	641109.0	9.562631e-01	0.187303	0.000000	1.000000	1.000000	1.000000	1.000000
NoDocbcCost	641109.0	9.731619e-02	0.270049	0.000000	0.000000	0.000000	0.000000	1.000000
DiffWalk	641109.0	2.517780e-01	0.407650	0.000000	0.000000	0.000000	0.514000	1.000000
Sex	641109.0	4.490697e-01	0.465399	0.000000	0.000000	0.224309	1.000000	1.000000
BMI	641109.0	1.727574e-13	1.000001	-5.684790	-0.604742	0.002460	0.636123	3.977620
PoorGenHlth	641109.0	-3.028245e-14	1.000001	-1.903140	-0.828289	0.153118	0.797146	1.939680
MentHlthProbs	641109.0	-1.500797e-13	1.000001	-0.693368	-0.693368	-0.693368	1.179437	1.725113
PhysHlthProbs	641109.0	4.101183e-13	1.000001	-0.848773	-0.848773	-0.848773	1.082332	1.548105
Age	641109.0	-1.607091e-13	1.000001	-2.264822	-0.737511	-0.001319	0.829535	1.748375
Education	641109.0	3.151464e-13	1.000001	-2.580164	-0.979582	-0.005239	1.218362	1.218362
Income	641109.0	2.086203e-13	1.000001	-1.847739	-0.851167	0.085771	1.050350	1.208134



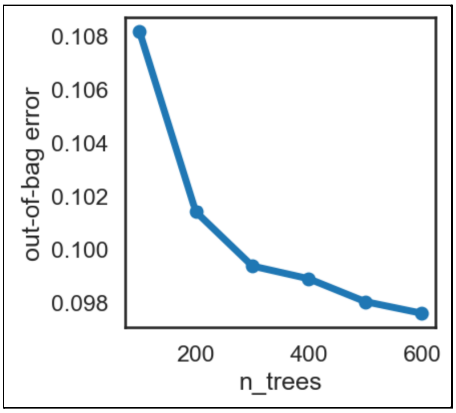
The first step in transforming the data was to make the continuous/scalar features closer to a normal distributions. All of them had some level of skew. The records that are on the far reaches of the long tails could have an outsize influence on the clustering which we would want to avoid. After using a Yeo-Johnson transformation and a Standard Scaler we see below the features have more of a normal distribution with fewer extreme values.



When we start fitting clustering algorithms to our data, we'll want to visualize in two dimensions how the clusters segment the data. In order to get a rough sense of the density contours of our data in two dimensions, I used a Principal Components Analysis (PCA) to reduce the data down to two principal components. I then created a kernel density plot to visualize the data (shown to the right). We see there is one main area of high density. This indicates that the data in its current state does not show multiple clusters. Therefore, the next step is to add weights to the features based on feature importance with respect to the classification feature. By prioritizing certain features over others, this will hopefully create a greater deal of segmentation in the data which will benefit the ability to use clustering algorithms later on.

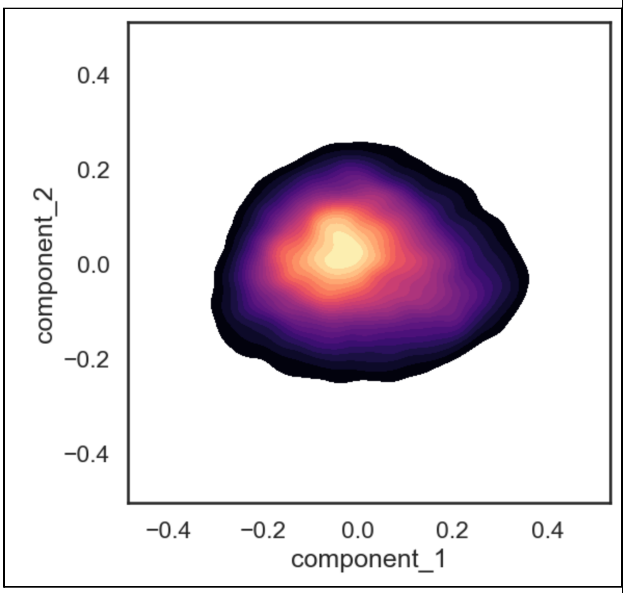


In order to derive feature importance, I decided to use a Random Forest model using the 'Diabetes_012' as the target classification feature. I fit the model multiple times using different numbers of trees to see the extent to which adding more trees reduces the out-of-bag error. Looking at the line graph to the right, we see there appears to be a large reduction in error when increasing the number of trees from 100 to 300. After 300 trees the error continues to decrease but starts to flatten out. Looking at the actual values of the out-of-bag error, we see that for this data, adding additional trees does not dramatically reduce the error. I, therefore, decided it best to use 300 trees. After fitting the model using 300 trees, I output the importances of each feature, as shown in the data frame to the right. We see the first six or seven features are the most important, and after that the importances start to flatten out.



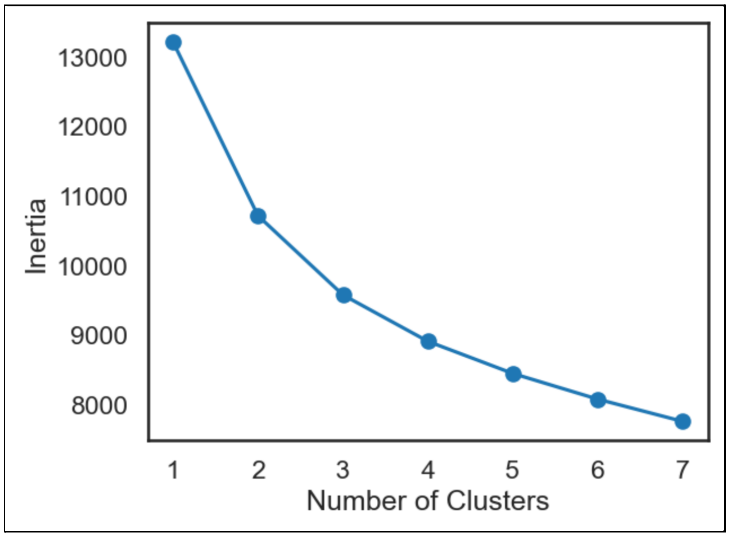
	feature	importances
0	BMI	0.108519
1	PoorGenHlth	0.101579
2	Age	0.09628
3	HighBP	0.087803
4	HighChol	0.078242
5	Income	0.075768
6	Education	0.062882
7	Smoker	0.045834
8	PhysHlthProbs	0.04512
9	Sex	0.044374
10	Fruits	0.039731
11	PhysActivity	0.037591
12	MentHlthProbs	0.036741
13	DiffWalk	0.032215

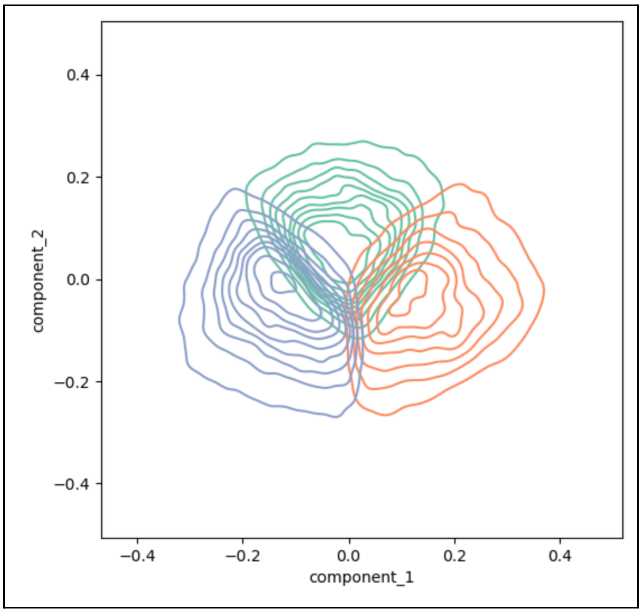
I then multiplied each feature in the data by its respective weight. After reducing the weighted data to two features I was able to create the kernel density plot to the right. Though much of the density is still congregated in the center, we now see slightly more variation in the contours and less round shapes. Therefore, there will not be a clean way to establish clusters with this data as the clusters will inevitably create cluster boundaries in high density areas.



K-Means Model

The first clustering algorithm I chose for this data was K-Means. After fitting models for all *k* between 1 and 7, I created the graph (shown below) of inertia per *k* to determine whether there is an “elbow” point that would indicate an ideal number of clusters to choose. Unfortunately the inertia decreases in the shape of a smooth exponential curve. As discussed above, there do not appear to be any naturally occurring clusters in the data that are both compact and isolated from other clusters, therefore there is no number of clusters we’re looking for that perfectly fit the data. To see how the algorithm chose to create the boundaries, I chose the model where k=3, and assigned the cluster labels to the kernel density plot used previously to see where the boundaries were drawn. Looking at the density plot below we see the algorithm segmented the data almost perfectly into three segments of equal size and shape.





The table below shows the percentage distribution of the three classes within each cluster. We see cluster 0 has a strong concentration in class 0 (i.e. no diabetes). Clusters 1 and 2 have a much stronger skew towards classes 1 and 2.

	0	1	2
Diabetes_012			
Diabetes_0_count	0.635717	0.145930	0.252735
Diabetes_1_count	0.232704	0.377635	0.376988
Diabetes_2_count	0.131579	0.476435	0.370277

The table below shows the most important features of the data and the extent to which they are above or below the mean value for each feature. The details are as follows:

- Cluster 0 is defined by below average values in for the features ‘BMI’, ‘PoorGenHlth’, ‘Age’, HighBP’ and ‘HighChol’, and above average values in ‘Income’ and ‘Education’.
- Cluster 1 is defined by high values in ‘BMI’, ‘PoorGenHlth’ and ‘HighBP’, and lower values for ‘Income’ and ‘Education’.
- Cluster 2 is defined by relatively low values for ‘BMI’ and ‘PoorGenHlth’ and high values for ‘Age’ and ‘HighBP’.

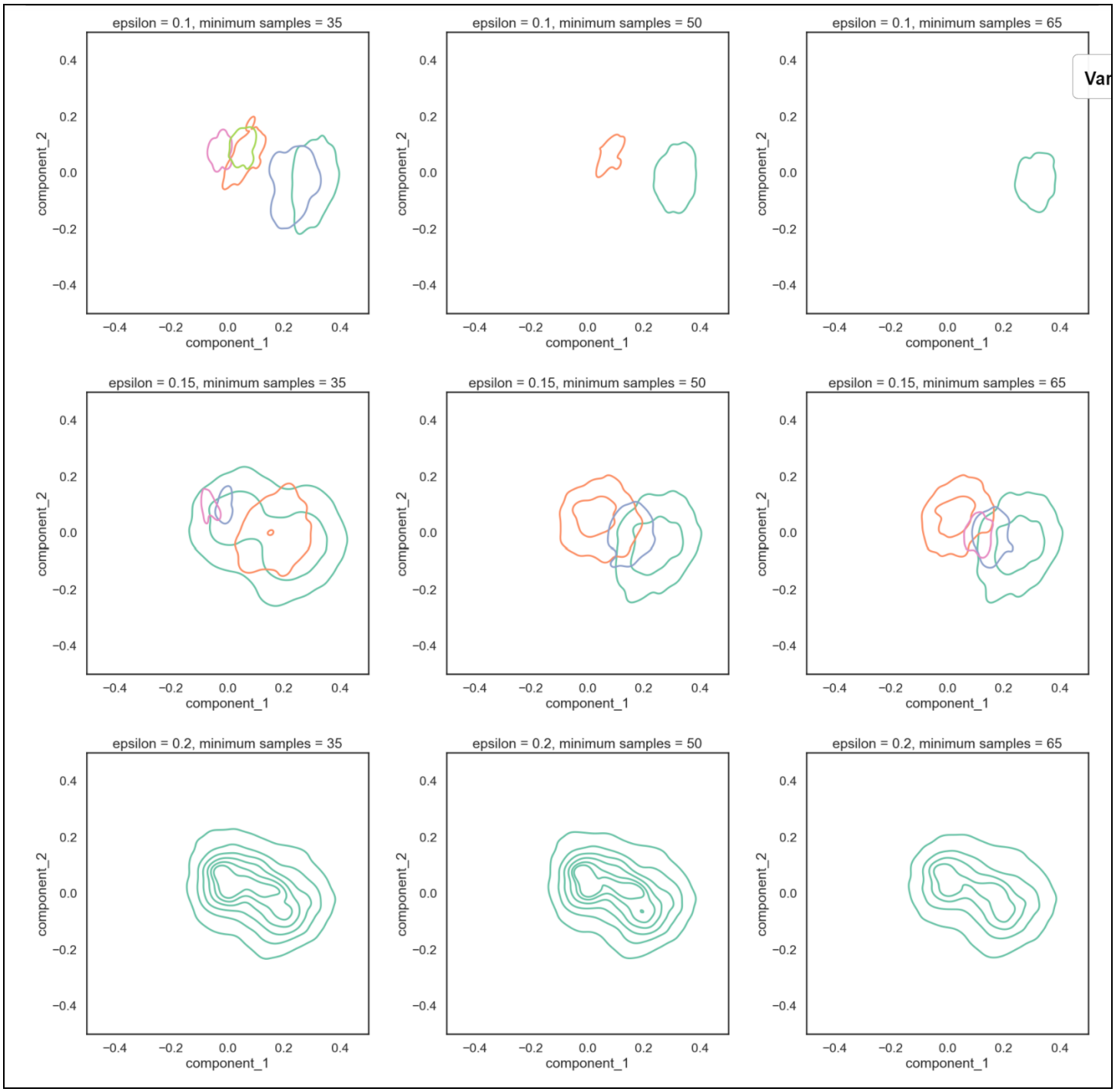
feature	0	1	2	importances
BMI	-0.052887	0.082011	-0.032157	0.108319
PoorGenHlth	-0.075079	0.088868	-0.019046	0.102185
Age	-0.065166	-0.017013	0.070648	0.096094
HighBP	-0.097934	0.037057	0.048544	0.089641
HighChol	-0.046012	0.020470	0.020472	0.077225
Income	0.038362	-0.041499	0.005862	0.075892
Education	0.023711	-0.026376	0.004345	0.062401
Smoker	-0.010066	0.006855	0.001971	0.045306
PhysHlthProbs	-0.017640	0.028653	-0.011735	0.045036
Sex	-0.000306	-0.005077	0.004883	0.044235

DBSCAN Model

The next clustering method I used was the DBSCAN algorithm. After multiple iterations, I found epsilon and minimum sample hyperparameters that produced workable clustering frameworks. The epsilon values chosen were 0.1, 0.15 and 02. The minimum samples values chosen were 35, 50 and 65. After fitting the models on the different combinations of hyperparameters I output a table showing what percentage of all records are grouped into each cluster (see below). It is clear from the table that for all models using epsilon=0.1 or 0.15, over 87% of the records are labeled as -1, meaning they are unclustered noise. The models using epsilon=0.2 have between 69% and 77% labeled as -1, which is an improvement over the others, however the algorithm could only find one cluster with this epsilon.

	0.1_35	0.1_50	0.1_65	0.15_35	0.15_50	0.15_65	0.2_35	0.2_50	0.2_65
-1	0.976	0.993	0.996	0.875	0.907	0.931	0.691	0.731	0.768
0	0.010	0.006	0.004	0.109	0.052	0.042	0.309	0.269	0.232
1	0.003	0.001	0.000	0.015	0.033	0.022	0.000	0.000	0.000
2	0.008	0.000	0.000	0.001	0.008	0.004	0.000	0.000	0.000
3	0.001	0.000	0.000	0.001	0.000	0.002	0.000	0.000	0.000
4	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Below are the kernel density plots for all nine combinations with records labeled -1 removed. We graphically see what was discussed previously: an epsilon of 0.1 or 0.15 finds a handful of non-dense clusters, but the majority of the records are considered noise. When epsilon=0.2, the models find one larger cluster.



I chose to examine further the model using epsilon=0.15 and minimum samples=35, because it has more than one cluster and has the lowest percentage of records in the -1 cluster of all the models where epsilon is either 0.1 or 0.15. The table to the right shows the distribution of records in each of the 'Diabetes_012' classes for each cluster. We see that clusters 0 and 1 have a high percentage of records in class 0, which refers to individuals without diabetes. For clusters 1 and 2, there are relatively less records in class 0 and more in classes 1 and 2, referring to pre-diabetes and diabetes.

	-1	0	1	2	3
Diabetes_0_count	0.297128	0.585629	0.654122	0.166667	0.103448
Diabetes_1_count	0.349788	0.220838	0.164875	0.500000	0.448276
Diabetes_2_count	0.353084	0.193533	0.181004	0.333333	0.448276

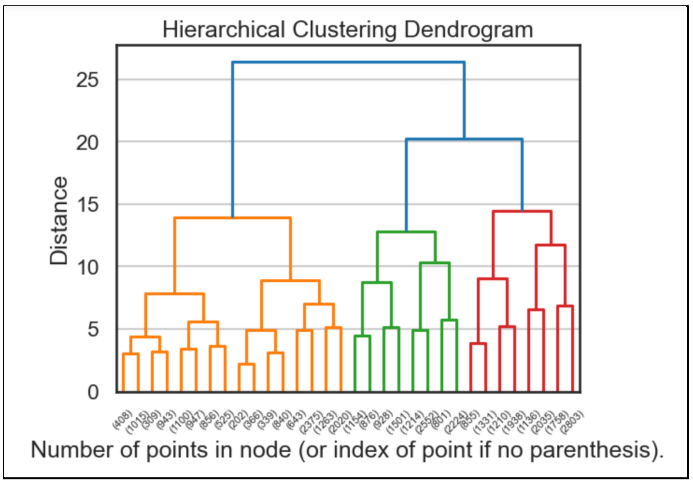
Looking closer at the distribution of the features among the different clusters (shown below), we can identify the features that make each cluster unique.

- Cluster 0 is defined most by 'PoorGenHlth' being below average (i.e. better general health), and 'Income' and 'Education' being above average.
- Cluster 1 is defined most by 'PoorGenHlth' and 'HighBP' being below average, while 'HighChol', 'Income' and 'Education' are above average.
- Clusters 2 and 3 are similar in that they are both defined by 'Age', 'HighBP', and 'HighChol' being above average and 'Education' being below average. Cluster 2 also has 'PoorGenHlth' being below average. Cluster 3 has 'Smoker' being above average.

	-1	0	1	2	3	importances
feature						
BMI	0.006513	-0.048866	-0.048026	-0.006021	-0.020187	0.108319
PoorGenHlth	0.012890	-0.091303	-0.096771	-0.082128	0.010698	0.102185
Age	0.003004	-0.027202	-0.018877	0.077238	0.067056	0.096094
HighBP	0.005592	-0.031808	-0.113321	0.075578	0.078466	0.089641
HighChol	0.003805	-0.037048	0.070985	0.069306	0.071840	0.077225
Income	-0.010527	0.072477	0.079330	-0.024388	-0.008032	0.075892
Education	-0.008633	0.058606	0.069652	-0.057386	-0.057789	0.062401
Smoker	0.003332	-0.020219	-0.027766	-0.039581	0.048678	0.045306
PhysHlthProbs	0.004571	-0.032183	-0.032547	-0.038198	-0.032683	0.045036

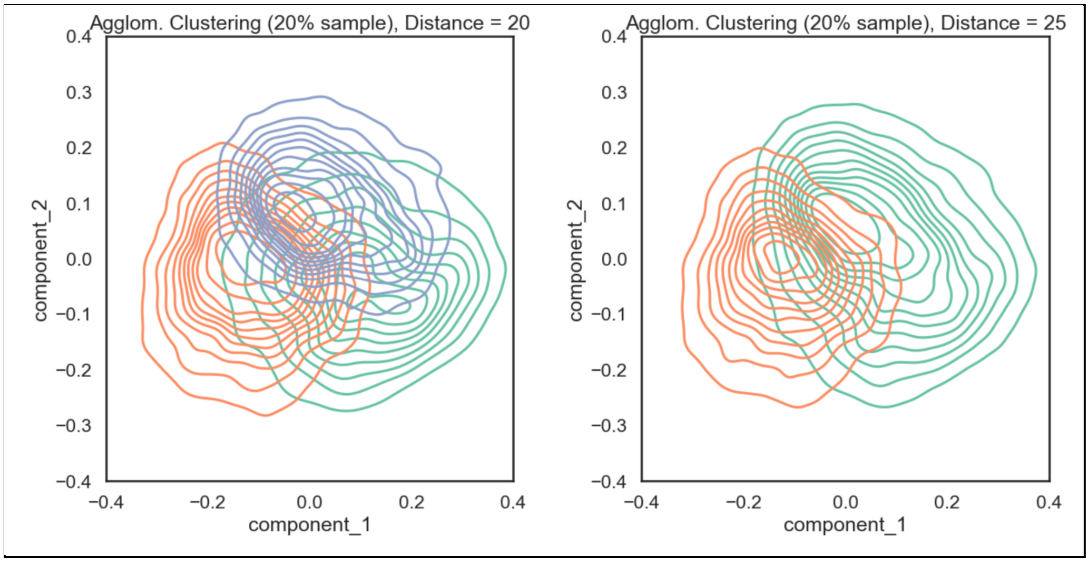
Agglomerative Hierarchical Model

For the Agglomerative Hierarchical model I downsampled the data to about 40,000 records as the computing time on my machine explodes beyond that number of records using an Agglomerative model. I did, however, maintain an even concentration of the three 'Diabetes_012' classes. After fitting the data using Ward linkage, I was able to output the dendrogram to the right. Looking at the dendrogram, it appears the appropriate distance threshold would be either 20 or 25, which would yield 2 or 3 clusters respectively.



After refitting the model using distance thresholds of 20 and 25, we can output a table showing the percent of records in each cluster. We also can output kernel density plots showing how the clusters map to our two dimensional view of the data. We see in the kernel density plots that unlike the previous two models, these models have far more overlap between the clusters. Given that the original kernel density plot did not show highly segregated clustering, this model that has overlap between clusters appears to better fit the contours of the data compared to the K-means model which had abrupt boundaries between the clusters. When the distance threshold is set to 20, three clusters are created that respectively constitute 34%, 37% and 29% of the data. When the threshold is increased to 25, clusters 0 and 2 are combined into the new cluster 0 which constitutes 63% of the data. I chose to move forward with the threshold set to 20.

	20	25
0	0.340	0.632
1	0.368	0.368
2	0.292	0.000



I continued examining the results using the three clusters derived from the distance threshold being set to 20. The table to the right shows what percentage of records in each cluster falls into the three 'Diabetes_012' classes. It is clear that clusters 0 and 1 are somewhat mirror images of each other, where cluster 0 has relatively high concentration of records in class 0 (no diabetes), and cluster 1 has relatively high concentrations in class 2. Cluster 2 has its class membership more evenly distributed among the three classes, but with the highest concentration in class 1.

	0	1	2
Diabetes_012			
Diabetes_0_count	0.522578	0.160130	0.331467
Diabetes_1_count	0.291903	0.352767	0.356978
Diabetes_2_count	0.185520	0.487103	0.311556

In the more detailed view of the features to the right, we see the following:

- Cluster 0 has relatively low values for 'BMI', 'PoorGenHlth', 'Age' and 'HighBP'.
- Cluster 1 has high values for 'BMI', 'PoorGenHlth' and 'HighBP'
- Cluster 2 has low values for 'BMI' and 'PoorGenHlth', and high values for 'Age' and 'HighBP'.

	0	1	2	importances
feature				
BMI	-0.029789	0.075712	-0.061733	0.108319
PoorGenHlth	-0.037282	0.061879	-0.034878	0.102185
Age	-0.048081	-0.003777	0.058972	0.096094
HighBP	-0.104027	0.065094	0.038693	0.089641
HighChol	-0.018570	0.018175	0.000274	0.077225
Income	0.018953	-0.018652	0.000678	0.075892
Education	0.009886	-0.014395	0.005658	0.062401
Smoker	-0.004232	0.003579	0.001502	0.045306
PhysHlthProbs	-0.008411	0.019419	-0.014742	0.045036
Sex	0.000999	0.000520	0.001482	0.044235

Key Findings and Recommendations

The key findings from these models is that irrespective of the clustering algorithm, we always find a cluster of individuals that are younger, healthier than the other clusters and with higher income and education. These are the clusters that have the lowest occurrence of pre-diabetes or diabetes. In the k-means and agglomerative clustering models we also see similar attributes for the clusters that do have an elevated occurrence of pre-diabetes and diabetes. In both cases there is a cluster primarily identified as having below average income and education, and above average occurrence of high BMI, poor general health, and high blood pressure. Also

in both cases we see a third cluster which is defined by below average BMI and above average age and high blood pressure.

As discussed previously, the data does not present any naturally occurring clusters that are highly segregated. Therefore, we cannot base our model preference based on how cleanly individual clusters are identified. Given this difficulty, it appears the k-means or agglomerative model works best at describing the data. When looking at the cluster characteristics in both algorithms, we see all three clusters have characteristics that are unique from one another and also show a meaningful correlation to our classification feature. The DBSCAN algorithm is less advantageous. Depending on the hyperparameters chosen, the DBSCAN algorithm either detects one cluster that represents the most dense area of all the data, or it detects multiple clusters that collectively represent less than 10% of total records.

Next Steps

If we were to continue working to better cluster this data, I would probably want to introduce additional features. I would also want to examine how the existing variables are derived and decide if changes need to be made with how it is structured. For example, the highest value in the 'Income' feature represents $75,000+, and we see that the histogram for 'Income' shows this maximum value has the most records. We may want to increase that maximum value or eliminate it altogether to we have a distribution that better represents a normal distribution.

By making these changes, I would hope we would start to see more contours in the data that would better point to unique clusters.