Problem 2: Heart Diseases

Idea: Applying clustering , PCA and t-SNE on heart diseases data set

We will divide the problem into:

1. Standardizing the data
2. Applying KMeans on the original dataset
3. Feature Reduction via PCA
4. Applying KMeans to PCA principal components
5. Feature Reduction via t-SNE
6. Applying KMeans to t-SNE clusters

1- Strandardizing the data

- The standardization of data will ultimately bring all features to the same scale and bringing the mean to zero and the standard deviation to 1

2-Applying KMeans on the original dataset

- Utilitize Elbow Method to determine the optimal number of clusters KMeans should obtain. It seem 4 clusters would be best to select.
- Apply KMeans on the original dataset requesting 4 clusters. We achieved a silhouette score of 0.11 which is on the low end.
- Using PCA to reduce the dataset into 3 principal components we can plot the

KMeans derived clusters into 2D and 3D visuals. PCA visualizations tend to aggregate clusters around a central point which makes interpretation difficult but we can see clusters 1 and 3 to have some distinct structure compared to clusters 0 and 2. However, when we plot the clusters into a 3D space we can clearly distinct all 4 clusters.

3-Feature Reduction using PCA

- First, let’s determine what is the optimal number of principal components we need. By examining the amount of variance each principal component encompasses, we apply PCA again and reduce our dataset to 3 principal components.

4- Applying KMeans to PCA principle components

- Now that we have reduced the original dataset of 15 features to just 3 principal components let's apply the KMeans algorithm. We once again needed to determine what is the optimal number of clusters and again it seems 4 is the right choice. It is important to remember we are now using the 3 principal components instead of the original 15 features to determine the optimal number of clusters.
- Then apply KMeans on the PCA principal components. We can see that we were able to increase our silhouette score from 0.11 to 0.28 by passing KMeans a lower dimensional dataset. Looking at the 2D and 3D scatter plots we can see a significant improvement in the distinction between clusters.

5- Feature Reduction using t-SNE

- We can see a definite improvement in KMeans ability to cluster our data when we reduce the number of dimensions to 3 principal components. In this section we will reduce our data once again using t-SNE and compare KMeans results to that of PCA KMeans. We will reduce down to 3 t-SNE components. Please keep in mind t-SNE is a computationally heavy algorithm. Computational time can be reduced using the 'n_iter' parameter. Furthermore, the code you see below is a result of dozens iterations of the 'Perplexity' parameter. Anything above a perplexity of 80 tended to aggregate our data into one large disperse cluster.

6- Applying KMeans to t-SNE clusters

- It seems 4 is the number of clusters for our KMeans analysis.
- Applying KMeans to our 3 t-SNE derived components we were able to obtain a Silhouette score of 0.24.

Resources:

- https://www.analyticsvidhya.com/blog/2021/04/k-means-clustering-simplified-in-python/
- https://builtin.com/machine-learning/pca-in-python
- https://www.datatechnotes.com/2020/11/tsne-visualization-example-in-python.html#:~:text=We%27ll%20start%20by%20loading%20the%20required%20libraries%20and%20functions.&text=After%20loading%20the%20Iris%20dataset,label%20parts%20of%20the%20dataset.&text=Then%2C%20we%27ll%20define%20the,the%20number%20of%20target%20dimensions

Note: Data used in this question is data for classification that’s why results seem to be strange.