Machine learning Hw 5 report

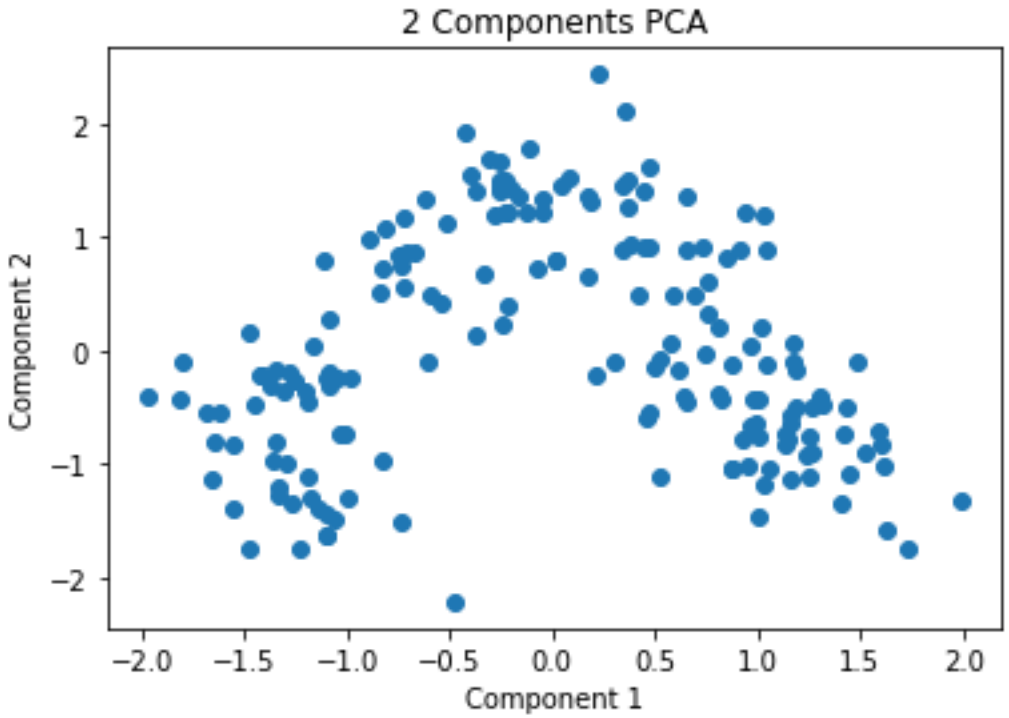
Jiajun Li
jl11895

Q1:

1. Do a PCA on the data. How many Eigenvalues are above 1? Plotting the 2D solution (projecting the data on the first 2 principal components), how much of the variance is explained by these two dimensions, and how would you interpret them?

   First of all, the data is normalized because different variables are measured by a different scale and if we do not normalize the dataset, large variables will dominate the process and leads to biased result. And a PCA model is built upon the normalized data with whiten equals true. Secondly, the eigenvalues can be easily calculated. There are three eigenvalues above one in this case. In this example, since we only need a 2D plot, only two principle components are extracted out and used as new axis to make the dimension reduced plot. Notice, the principle components are not chosen by random. Only the principle components associated with the highest and the second highest eigenvalues are extracted. The variance explained by each of the principle components is 36.2% and 19.21%.

   Here is how I interpret the result.



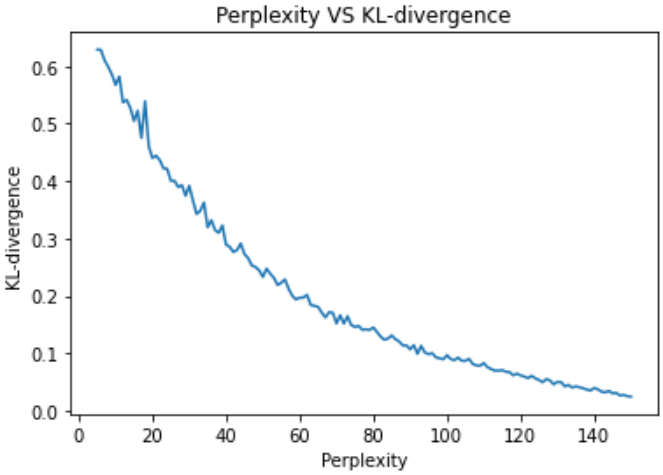
If you look at the graph, we can discern a clear parabola shape and in addition with the fact that we have three eigenvalues above in this case, I would say that there is a good chance that we could be three clusters in this dataset, which suggests that we have three different types of wine. Furthermore, the first two principle components, in total, explains about 55% of the overall variability of the data set. It is not a very decent percentage. I would say there maybe three dimension may exhibit a better local structure while preserving the global structure of the dataset.

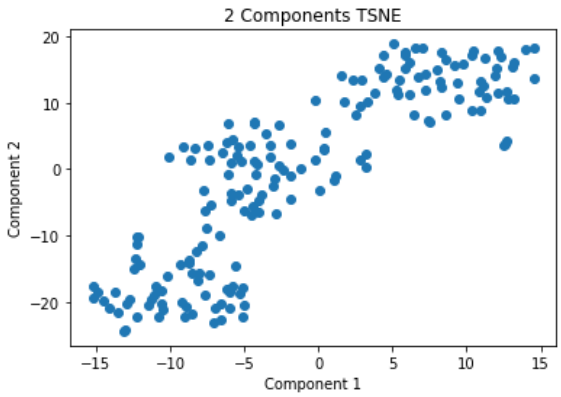
The number of Eigenvalues above 1 is 3

The variance explained by these three dimensions are 0.3619884809992631 and 0.19207490257008925 and 0.1112363053624998

Q2: Use t-SNE on the data. How does KL-divergence depend on Perplexity (vary Perplexity from 5 to 150)? Make sure to plot this relationship. Also, show a plot of the 2D component with a Perplexity of 20.

Again for the same reason, the dataset is normalized and then a tsne model is build based on different perplexity values ranging from 5 to 150. For the second question, I built a tsne model with 2 components for simple visualization and a perplexity of 20





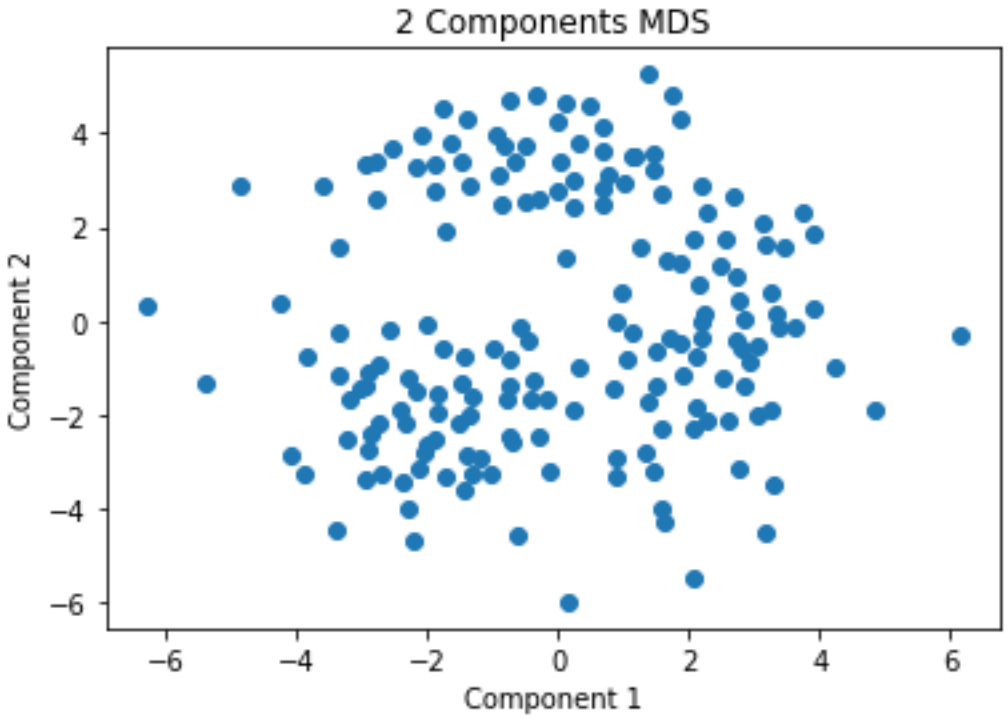
From the graph, we can easily discern the fact that the KL-divergence decreases as the perplexity increases. Even though lower KL-divergence means that the discrepancy between higher dimension and lower dimension is more minimized and we are delighted to

see a low KL-divergence. I tend not to draw the conclusion that this relationship is useful or benefits the interpretation of the model. In other words, I do not recommend to intentionally set a high perplexity number to lower the KL-divergence. The reason is that when calculating P(probability distribution) in the KL-divergence, higher perplexity broadens the Gaussian kernel that models the joint probability in the high-dimension space. The variance of the Gaussian increases leads to decrease in probability distribution and finally the KL-divergence. Thus, it is numerical that higher perplexity gives lower KL-divergence and one cannot use it to evaluate the performance of the model. It is better to the maintain the value of perplexity and change other hyperparameters. Thus for the second question, while maintaining the perplexity to 20, I adjust the number of iterations to 1200 and the final KL-divergence achieved is 0.4663. In addition, we

Q3: Use MDS on the data. Try a 2-dimensional embedding. What is the resulting stress of this embedding? Also, plot this solution and comment on how it compares to t-SNE.

For this question, the data is normalized and the MDS model is then performed. The build-in function mds.stress_ yields ridiculous stress value over 20000 so I decided to calculate the stress manually. First of all, the pairwised distance of the original dataset is calculated and arranged into a distance map. Then fit_transform the data into low-dimension points so the pair-wised distances in the low dimension can be calculated. Finally, the stress can be calculated using the stress formula. The stress gives us a quantitative measure of how well the model performs. And in this question, the stress value is 0.2258725, which is close to unfortunately not a really decent number. A better stress value should not exceed 0.2. In addition, I found out that if I increase the number of components to 3 instead of 2, the stress value reduced to 0.144179, which is wonderful. However, when we restrict to only 2

components, the stress value varies little even though I tried different hyperparameters.



Q4: Building on one of the dimensionality reduction methods above that yielded a 2D solution (1-3, your choice), use the Silhouette method to determine the optimal number of clusters and then use kMeans with that number (k) to produce a plot that represents each wine as a dot in a 2D space in the color of its cluster. What is the total sum of the distance of all points to their respective clusters centers, of this solution?

In this question, tsne is chosen as the dimension reduction method because from previous questions, it yields the best result. The tsne dimensionality reduction is built with perplexity 20.Then, I tried different number of clusters ranging from 2 to 9 and find the silhouette score. For the second part, I used the number of clusters with the best silhouette score to build a Kmeans clustering method. And finally, the distance of all the points to their respective center is summed up.
For n_clusters = 2 The average silhouette_score is : 0.58471423

For n_clusters = 3 The average silhouette_score is : 0.6187734

For n_clusters = 4 The average silhouette_score is : 0.526302

For n_clusters = 5 The average silhouette_score is : 0.46130526

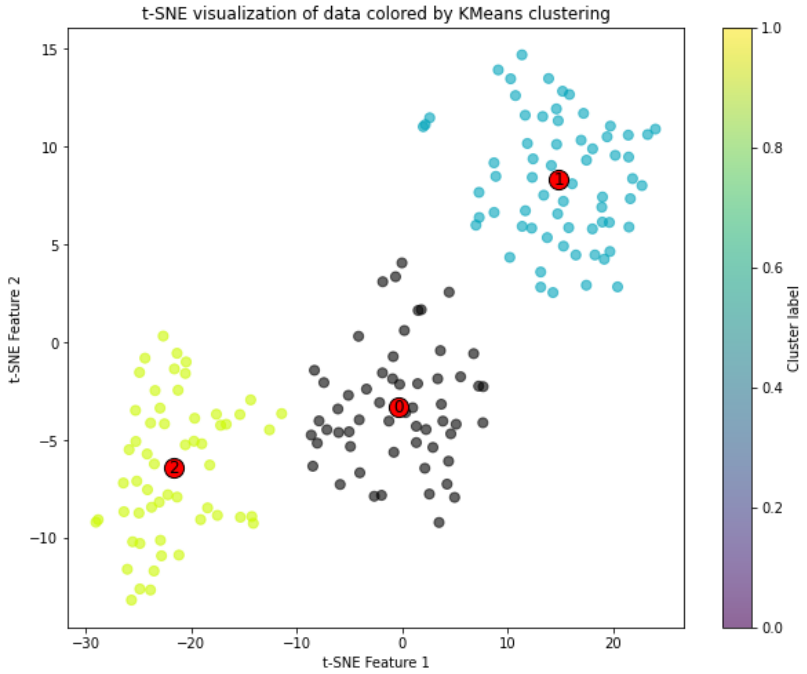
For n_clusters = 6 The average silhouette_score is : 0.4020541

For n_clusters = 7 The average silhouette_score is : 0.39217472

For n_clusters = 8 The average silhouette_score is : 0.40342617

For n_clusters = 9 The average silhouette_score is : 0.38484505

As you can see, the number of cluster with the best silhouette score is 3, which might suggest that we have 3 different types of wine in this dataset. And intuitively, for the second part, the Kmeans clustering method should have 3 clusters as higher silhouette indicates better Kmeans performance. The Kmeans algorithm can calculate the distance from each point to the cluster and classify each point based on the minimum distance. In this question, the distance of each point is the distance to its own cluster (the minimum distance of all the distances to three different clusters). The total sum is : The total sum of the distance of all points to their respective clusters centers is: 908.0809968709946



The graph using number of 3 clusters makes perfect sense.

Q5: Building on one of the dimensionality reduction methods above that yielded a 2D solution (1-3, your choice), use dBScan to produce a plot that represents each wine as a dot in a 2D space in the color of its cluster. Make sure to suitably pick the radius of the perimeter ("epsilon") and the minimal number of points within the perimeter to form a cluster ("minPoints") and comment on your choice of these two hyperparameters.

Certainly, the dataset is normalized for the same reason.

For this question, a tsne dimension reduction method is applied to generate a 2D projection with perplexity 20. The reason why tsne dimension reduction is chosen for this question is that not only it shows the best plot in 2D embedding based on the previous questions, but also it balances local and global structures.

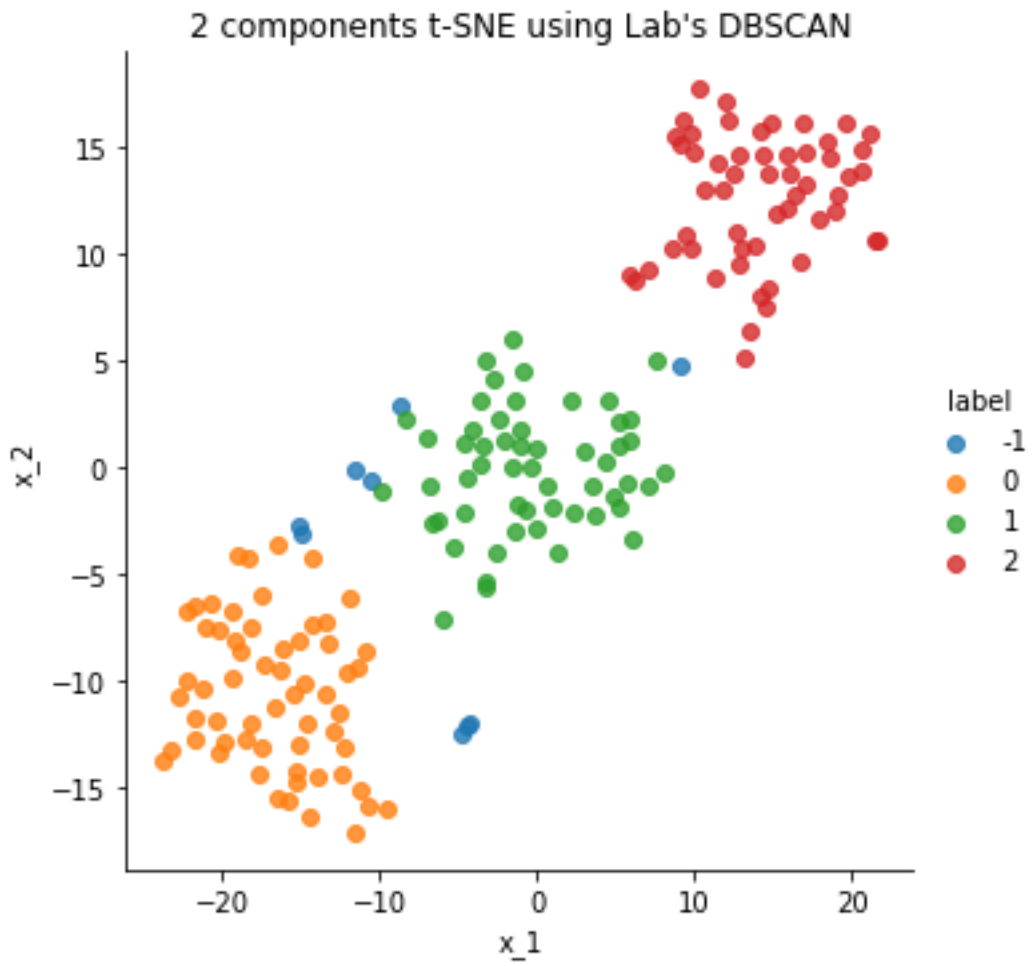
Then DBSCAN is applied to the embedded dataset and in this question, I kept trying different combinations of epsilon value and MinPoints to make a reasonable plot. Finally, epsilon value of 3.3 succeeds to classify nearly all the points into three clusters and a minimum point of 5 prevents few points to be classified as the 4th group and instead be classified as some noise.

Here is the result from the DBSCAN
Number of core points: 158

Number of clusters: 3

Number of unclassified points: 7

Here is what I found on the two hyperparameters epsilon and MinPoint. If keep Minpoint fixed, increasing the value of epsilon will make the whole plot tend to be just one cluster and decreasing the value of epsilon will increase both the number of clusters and the number of unclassified points. If I keep epsilon unchanged, increasing the number of Minpoint will increase the number of unclassified points and decrease the number of Minpoint will tend to make the plot into just one cluster. Following this property, I finally has the best result above with epsilon is 3.3 and Minpoint is 5. And the plot looks like this:



The plot and the DBSCAN concludes that it is reasonable to have 3 clusters in this case.

Extra A:

There are three types of wines. Based on Kmeans method and T-SNE plot, I would say that their difference could be very obvious. In other words, their differences are somewhat distinguishable.