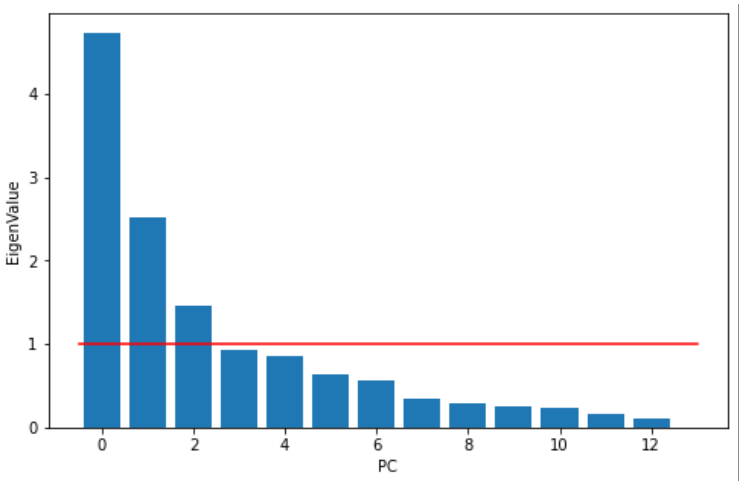
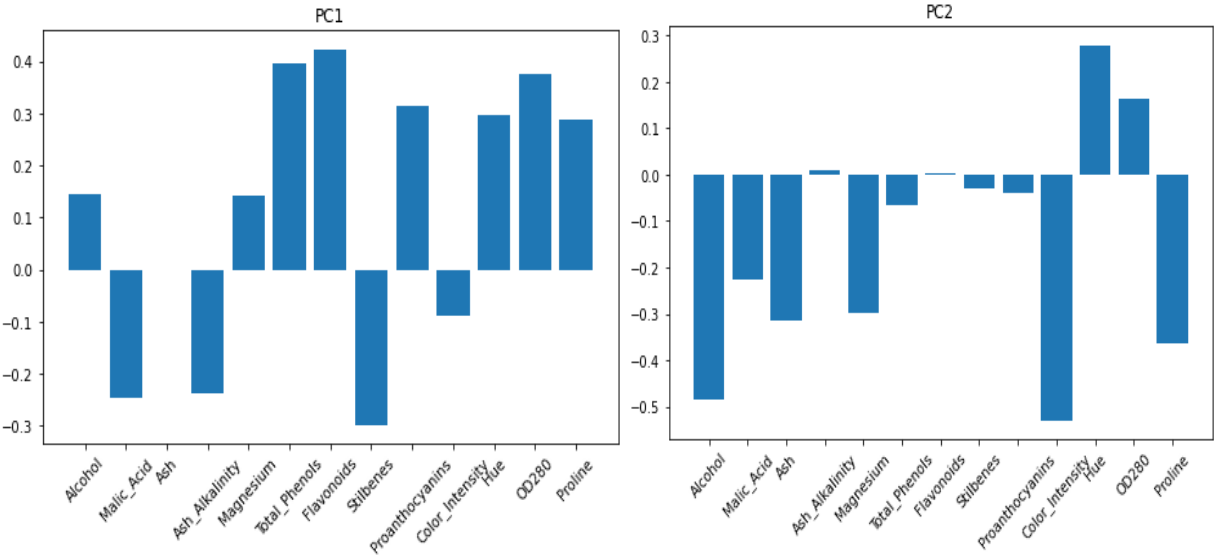
1. Do a PCA on the data. How many Eigenvalues are above 1? Plotting the 2D solution (projecting the data on the first 2 principal components), how much of the variance is explained by these two dimensions, and how would you interpret them?

To perform PCA, first I standardized the dataset to put all the features on the same scale so that all features have equal weight, and then fit it to Sklearn's PCA. To find the eigenvalues I used pca.explaine_variance, and made a bar graph (screeplot):

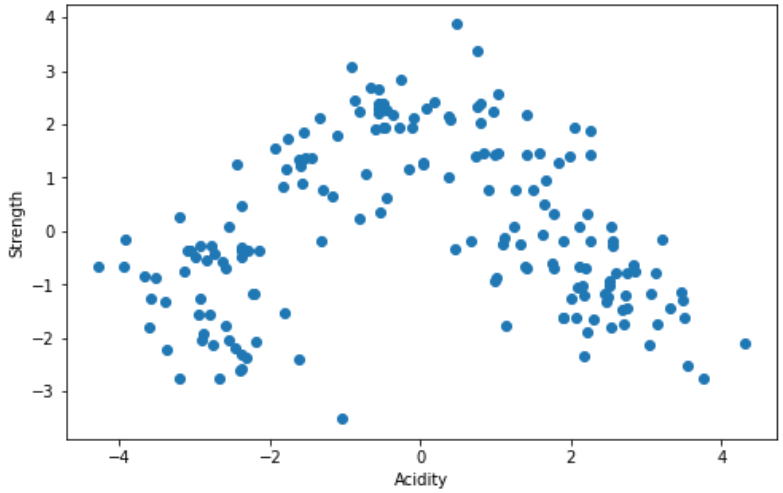


As demonstrated in the plot, 3 eigenvalues are above 1.
To see how much variance the first two principal components explain, I used pca.explained_variance_ratio and found that they explain 55.406% of the variance.
To interpret the first two PCs, I looked at the loadings matrix, which shows the correlation between each feature with the principal components that represent how much each feature "contributes" to the PC. Then, I made a bar graph of the absolute value of each feature's correlation with the principal components. The result for the first two PCs in order is below:



Total_phenols, flavonoids, and OD280 have high correlation with PC1. Through some brief research, I found the first two are acidic, and protein, which OD280 measures, is also acidic. So PC1 can be interpreted as acidity.

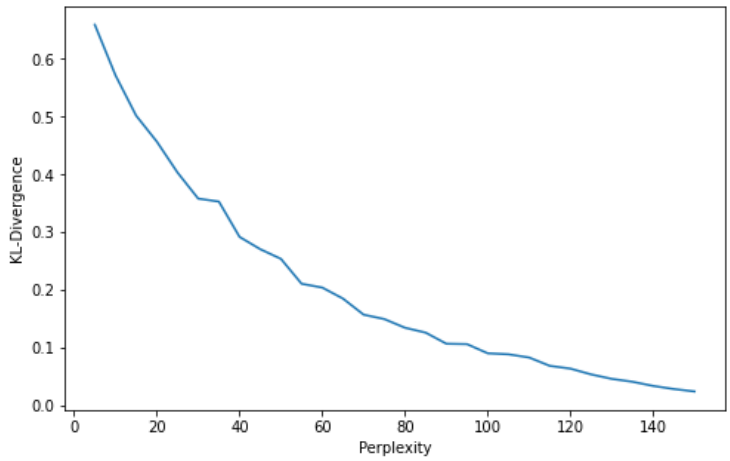
Color_Intensity and Alcohol have high correlation with PC2. It can be interpreted as the strength of the wine.
The scatter plot of the first two PCs is below



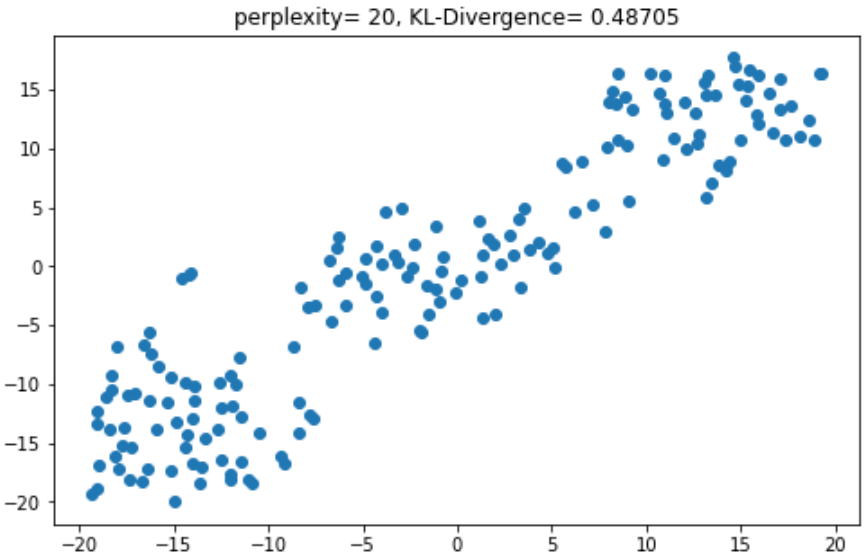
Just by looking at the plot, there might be 3 clusters (with a decent amount of noise). This will be tested in later questions.

2. 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.

To answer the first part of the question, I tested the KL-divergence with perplexity from 5 to 150 with an increment of 5 each step. The perplexity vs. KL-divergence plot is below”



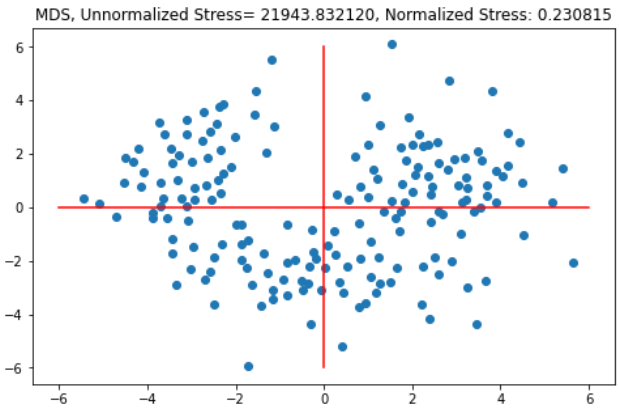
As demonstrated in the plot, as perplexity increases, KL-divergence decreases, but at a decreasing rate.
2D component with a perplexity of 20 is shown below:



This plot, compared to PCA, much more clearly shows the cluster of the dataset. It is more clear that there are 3 clusters.

3. 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.

First, I computed the distance matrix by using sklearn's euclidean_distances class. Then I fit the matrix to MDS with 2 components. The stress_ attribute, according to Sklearn's documentation, is the sum of squared distance between actual distance and predicted distance. To find normalized stress, divide the output of stress_ attribute by the total sum of squared distance multiplied by 0.5, and then square root the whole thing (`np.sqrt(stress / (0.5*np.sum(d ** 2)))`. "d" is the distance matrix. Unnormalized stress was 21943.832, normalized stress was 0.231. Because a stress value >0.2 is considered poor, the clustering performance is not good. The resulting scatter plot below supports this point:

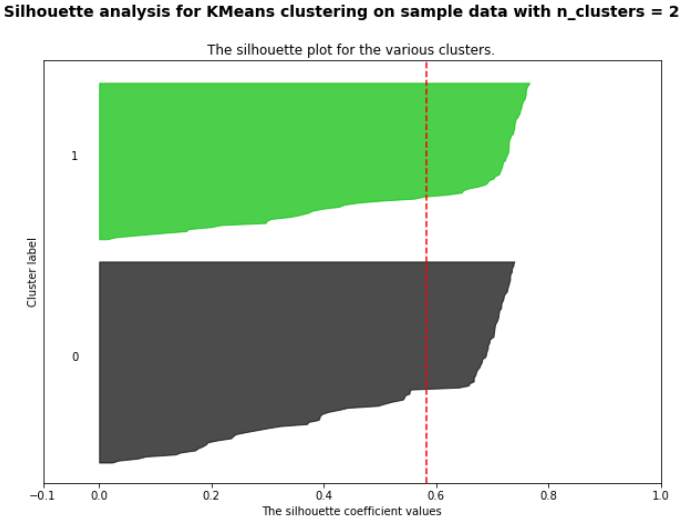


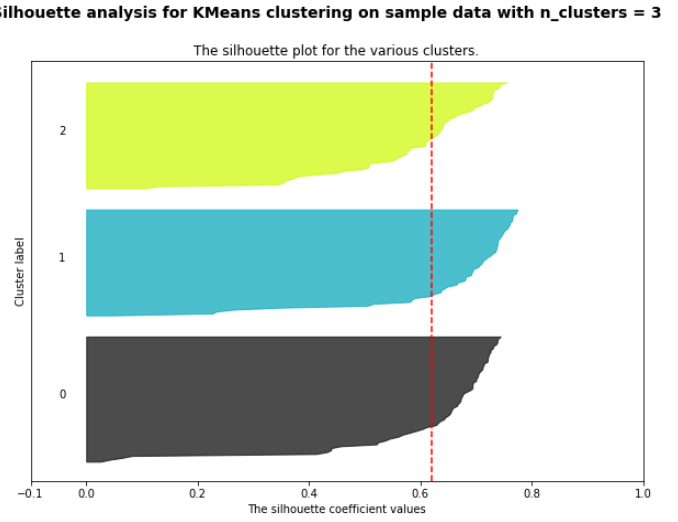
In comparison with t-SNE, the result of MDS transformation was not as good. T-SNE did a much better job in visualizing the possible clusters within the data. It is quite clear from t-SNE that the

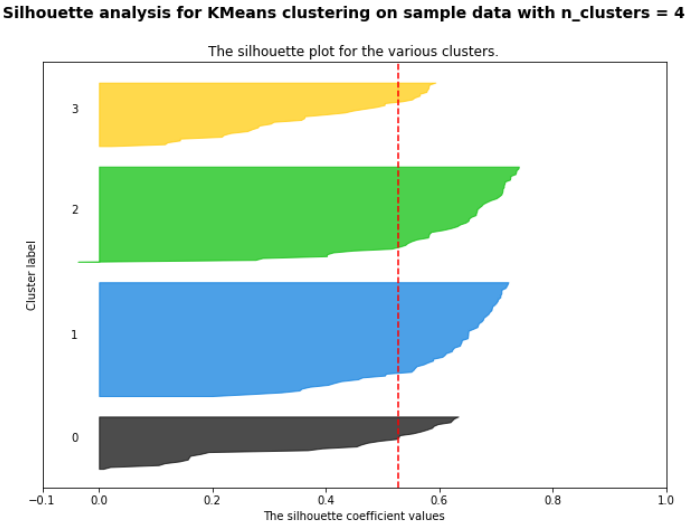
wines can be classified into 3 groups, while this is not the case for MDS. In the MDS plot, there are no distinguishable clusters. No data points seem to focus around the same point.

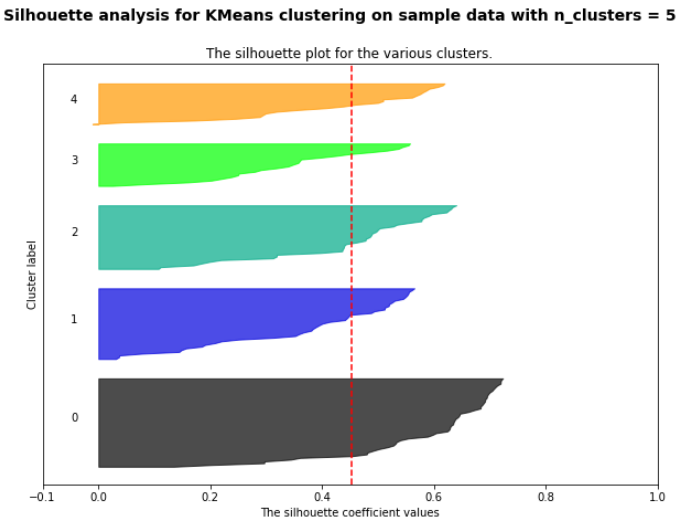
4. Building on one of the dimensionality reduction methods above that yielded a 2D solution (1-3, your choice), use the Silhouette method to determine the optimal number of clusters and then use kMeans with that number (k) to produce a plot that represents each wine as a dot in a 2D space in the color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

For this question, I chose t-SNE. Using code provided in the lab to plot Silhouette scores, I found that the optimal number of clusters is 3. The plots of clusters from 2 - 5 is below:



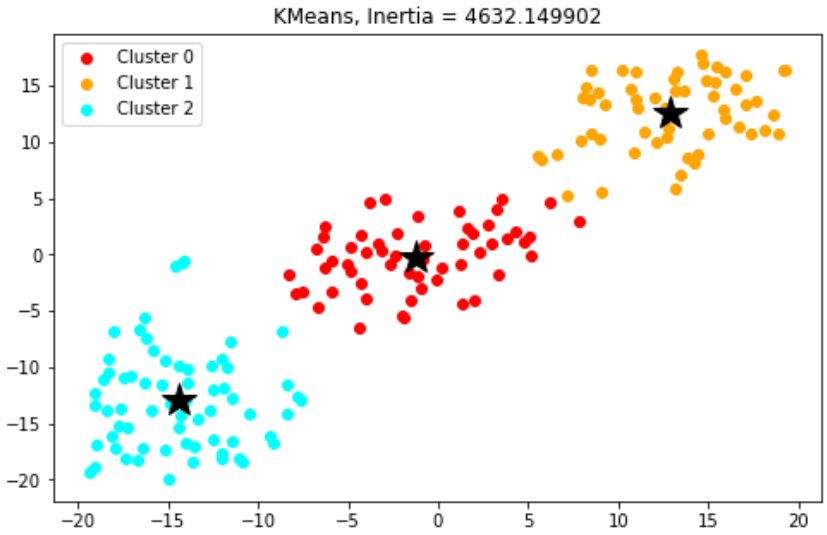






Cluster = 3 not only has the highest mean silhouette coefficient but the clusters also have very consistent silhouette coefficients. Therefore, 3 clusters is optimal.

The result of KMeans clustering with 3 clusters is below:

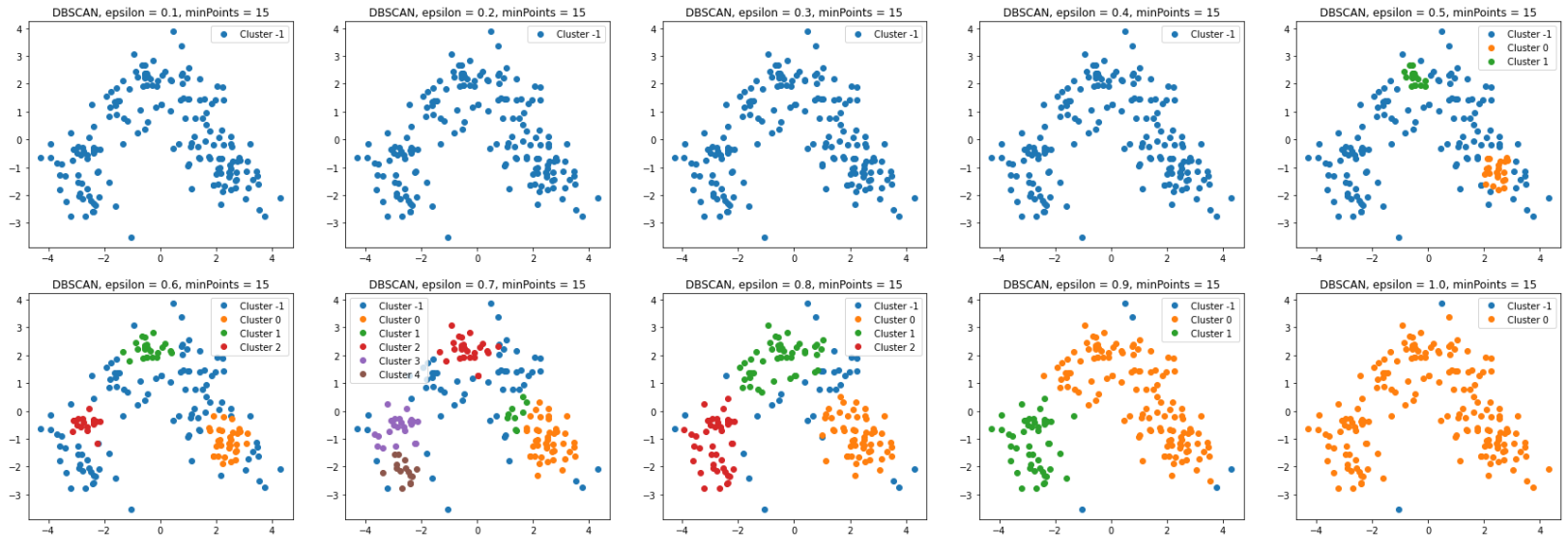


The black stars represent cluster centers.
According to Sklearn's documentation, the inertia_ attribute is the "sum of squared distances of samples to their closest cluster center." Using this attribute, I found that the total sum of the distance of all points to their respective clusters centers of this solution is 4632.150.

The fact that there are 3 clusters suggests that there are 3 types of wines.

5. 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

For this question, I chose PCA. To decide what radius/epsilon and perimeter/minPoints, I plotted all combinations of epsilon in {0.1, 0.2, … , 1} and minPoints in {5, 6, …, 15}. To save space, I will only show the graphs that contains the best result:

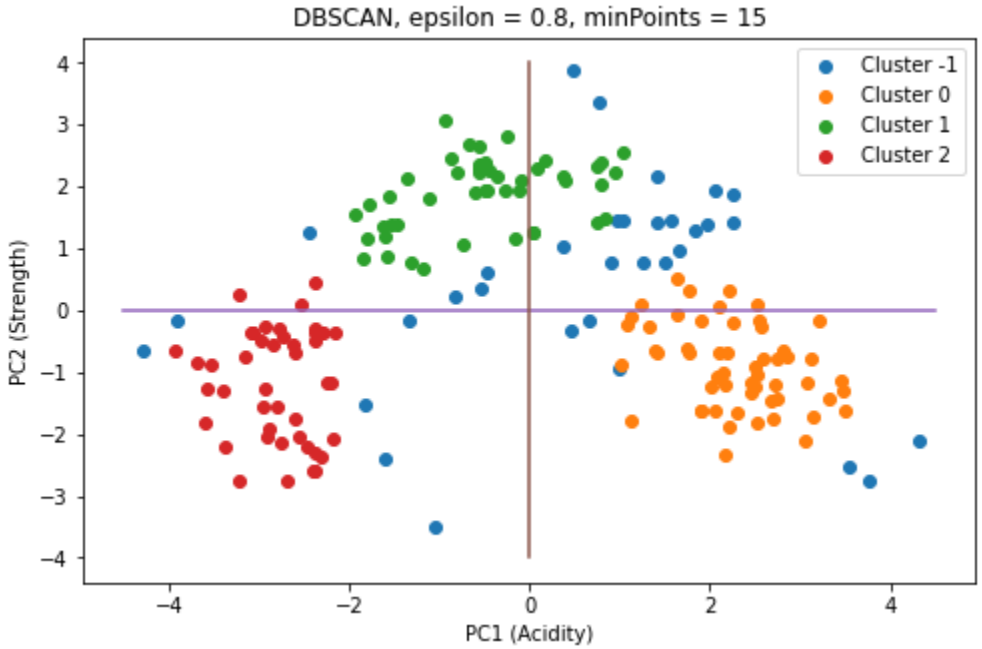


Epsilon = 0.8 and minPoints = 15 produced the best result (second row, third plot from left to right). As we can see in the plot, this set of hyperparameters give the best cluster with the least noise and more separability. DBSCAN with epsilon = 0.8 and minPoints = 15 produced 3 clusters. And from the plot, we can see that the clusters are very distinguishable, although there are a few unclassified points (blue).

Extra credit:

a) Given your answers to all of these questions taken together, how many different kinds of wine do you think there are and how do they differ?

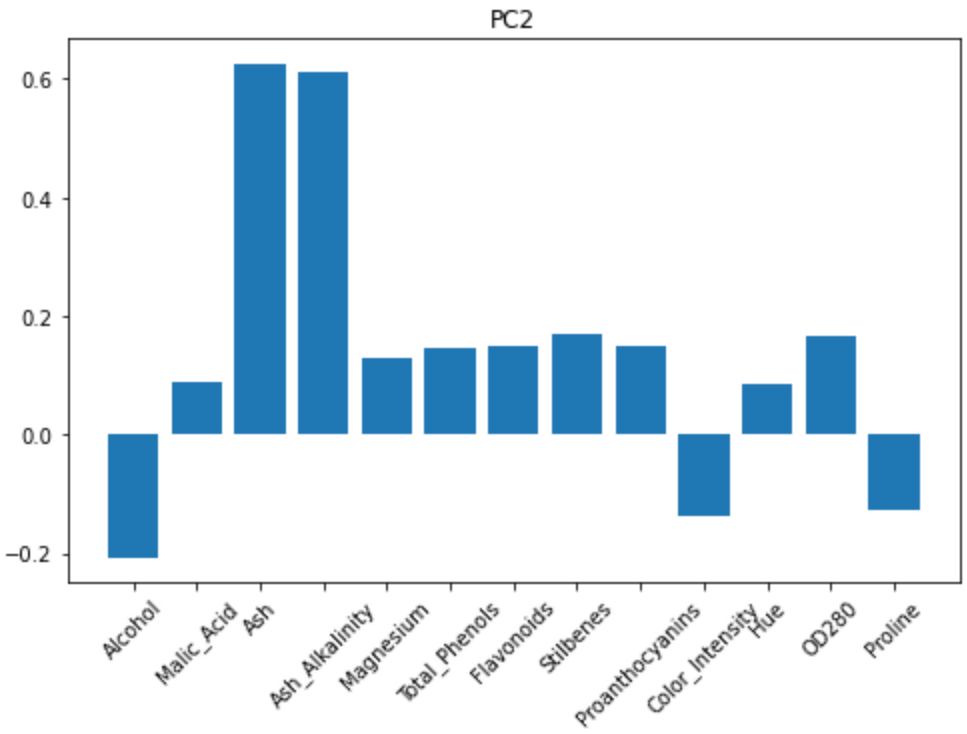
DBSCAN and Kmeans both indicate that there are 3 clusters, which means that there are 3 different kinds of wine. To answer how they differ, I examine the cluster on PCA:



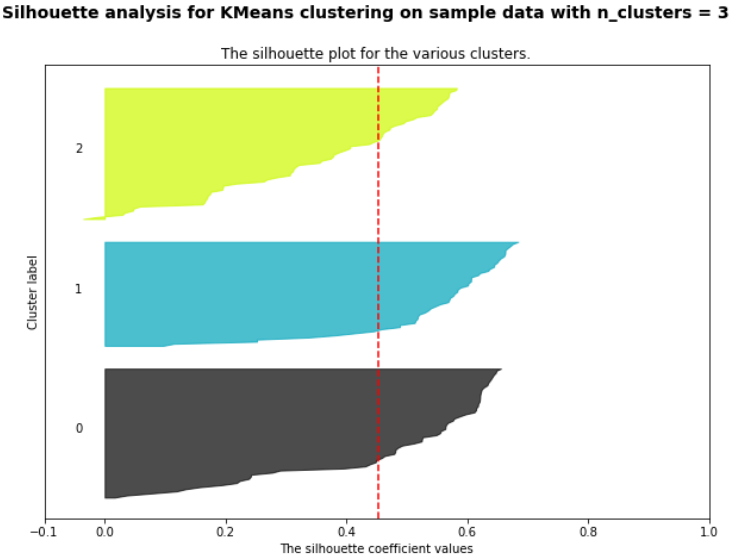
Cluster -1 means unclustered points. Cluster 0 (orange) is higher on acidity and lower on strength. Cluster 1 (green) is high on strength and average on acidity. Cluster 2 (red) is lower on acidity and lower on strength.
The three possible types of wine are: strong but moderately acidic, weak but highly acidic, and weak but low acidity.

b) Is there anything of interest you learned about wines from exploring this dataset with unsupervised machine learning method that is worth noting and not already covered in the questions above?

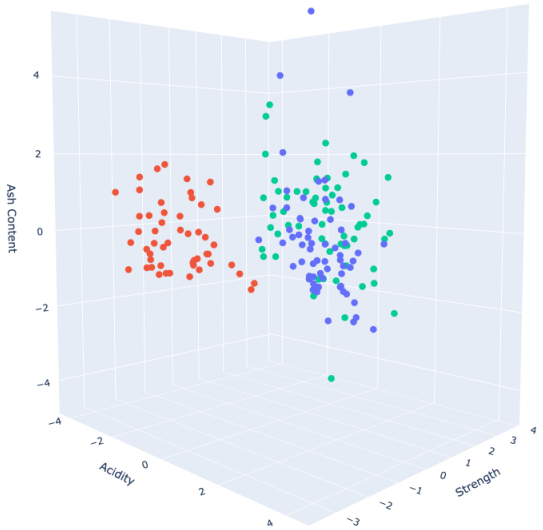
The loadings of the third principal component of the dataset is below:



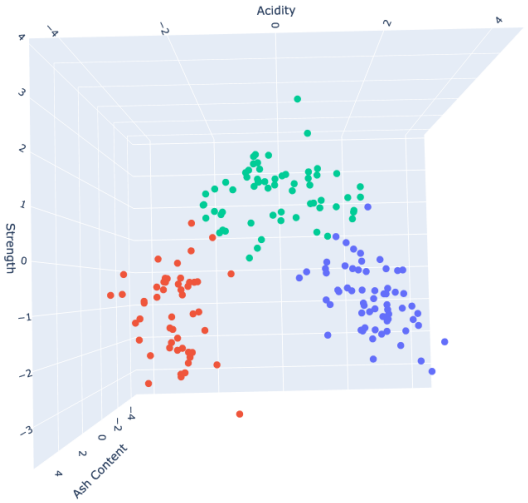
As demonstrated in this plot, Ash and Ash_Alkalinity are very highly correlated with this PC. These two features can be summarized by “Ash Content.” Using the same silhouette method in question 4, there are also 3 clusters. But because the silhouette coefficient is lower, it is anticipated that the cluster will not be as good.



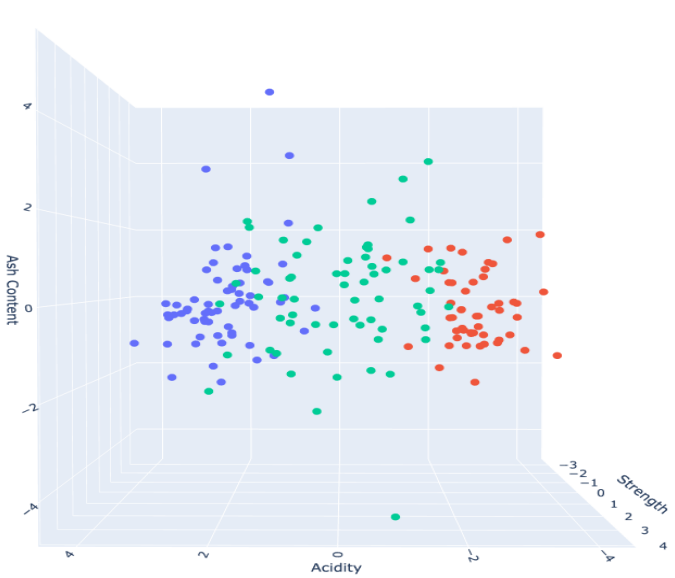
The 3d cluster is shown below (in the code file, this plot can be rotated):

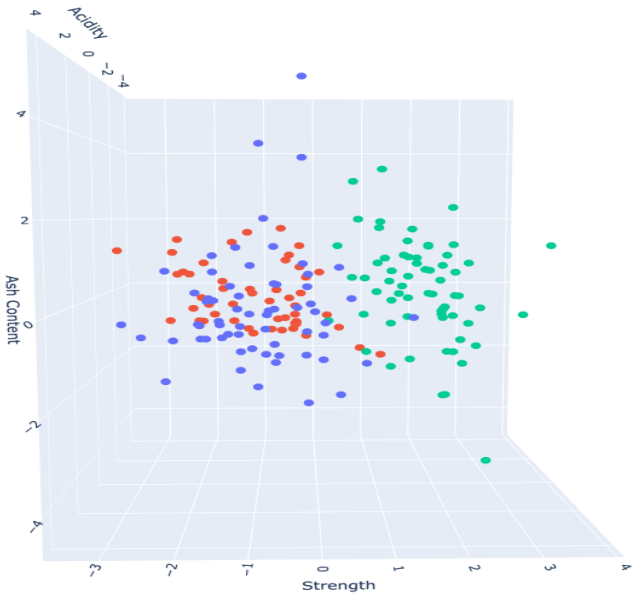


If we look at only at acidity vs. strength, we see a clear cluster:



But looking at ash content vs. any of the other two, this is not the case:





This suggests that acidity and strength are the most relevant factors in separating a class/type of wine from others.