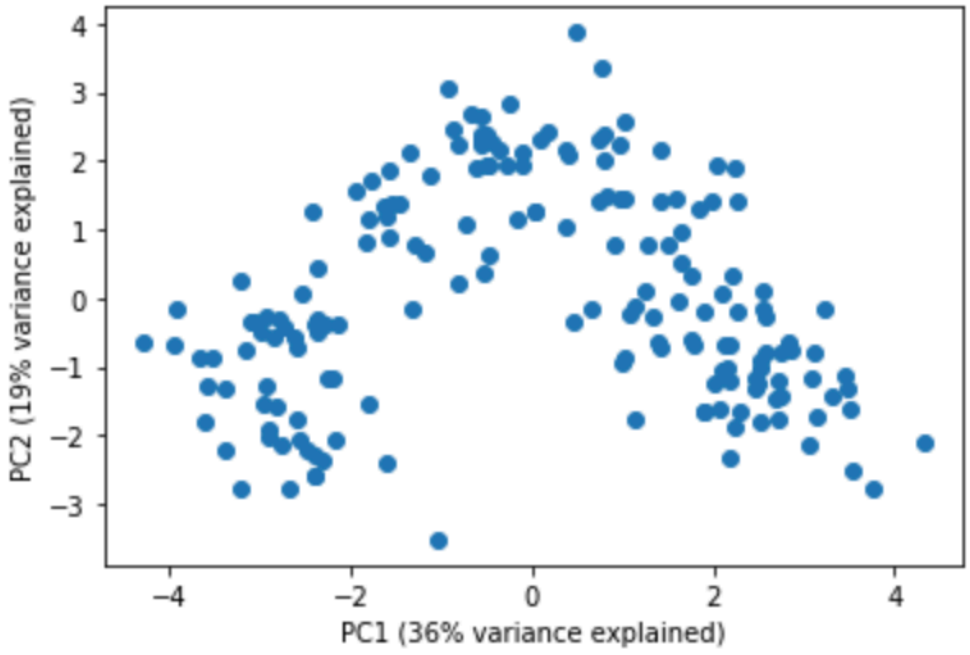
Data Preprocessing

As indicated that the data has been carefully curated and no missing values exist, it is validated by running data.info() function. Since we are attempting to determine how many different kinds of wine there are, the 13 columns in the dataset are all predictor variables that will be considered to categorize different kinds of wine. However, all the 13 variables are varying from each other too much. Therefore, I use standard scaler to scale it to normalize all the data in the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Alcohol          178 non-null    float64
 1   Malic_Acid       178 non-null    float64
 2   Ash              178 non-null    float64
 3   Ash_Alkalinity   178 non-null    float64
 4   Magnesium        178 non-null    int64  
 5   Total_Phenols    178 non-null    float64
 6   Flavonoids       178 non-null    float64
 7   Stilbenes        178 non-null    float64
 8   Proanthocyanins  178 non-null    float64
 9   Color_Intensity  178 non-null    float64
 10  Hue              178 non-null    float64
 11  OD280            178 non-null    float64
 12  Proline          178 non-null    int64  
dtypes: float64(11), int64(2)
memory usage: 18.2 KB
```

Question 1

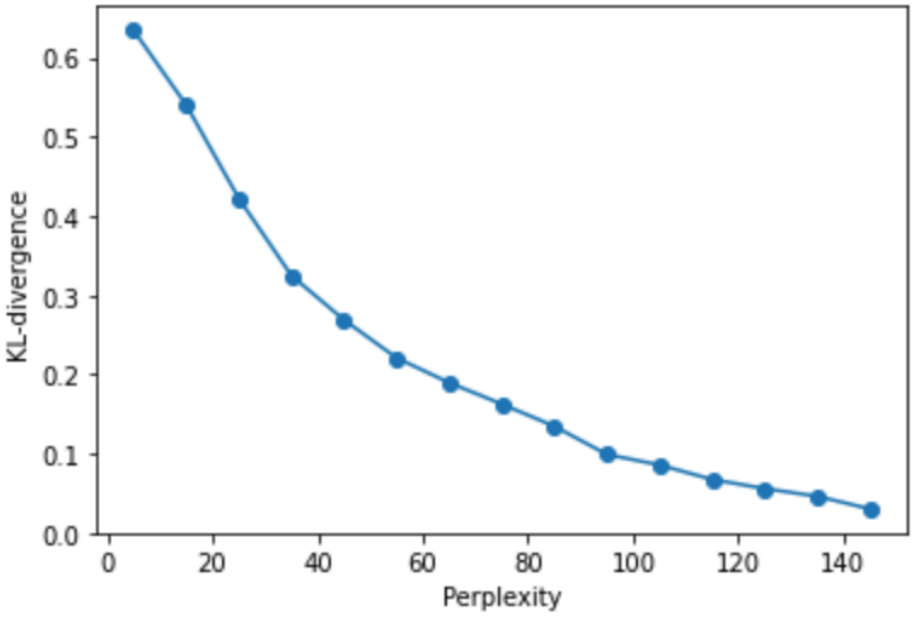
To determine how many eigenvalues are above 1, we have to find the covariance matrix, C using np.cov() then compute the k eigenvector of C with np.linalg.eig() function. Then I do a PCA for the data with n_components=2, fit_transform the normalized X in the data preprocessing process, plot the graph between the first 2 principal components and calculate the variance explained by these 2 dimensions with the explained_variance_ratio_[:2] function. PCA's goal is to reduce the number of prediction variables in the dataset while retaining the maximum amount of information. Running PCA with n_components = 2 will result in 2 principal components that can be used for 2D visualization. There are 3 eigenvalues that are greater than 1 where the first 2 higher valued eigenvalues are the 2 principal components used in the PCA plot. The total variance explained by the 2 principal components is 0.55406, equivalent to 55.406% where the first component explained approximately 36% of the variance and the second component explained approximately 19% of the variance. Below is the plot:



The total variance explained by the 2 principal components are the percentage of the total variance in the original dataset which indicates that the 2 principal component explained approximately 55% of the total variance in the original dataset. This number is quite high implying that the remaining 11 features are accounting for less than half of total variance. This suggests that even a few features is enough to describe the difference between wines and form clusters.

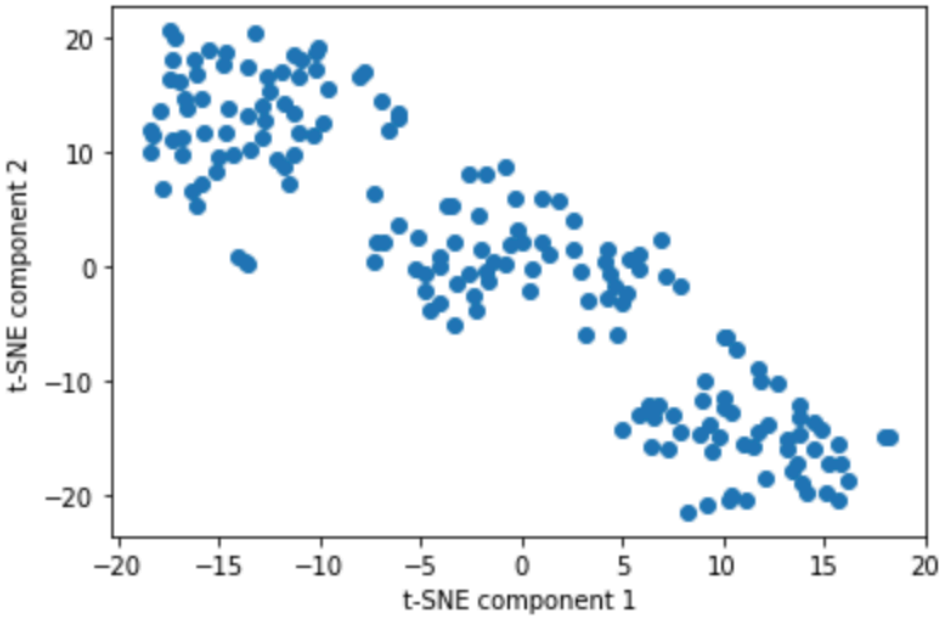
Question 2

I determine the relationship between KL-divergence and Perplexity varying from 5 to 150 with an interval of 10 and plot a graph of their relationship. t-SNE uses perplexity to balance the attention given to local and global aspects of the data while KL-divergence measures the difference between the distributions of the original high-dimensional space and the low-dimensional space. Below is the relationship plot:



The plot indicates that there is a negative relationship between perplexity and KL-divergence. When perplexity increases, KL-divergence decreases.

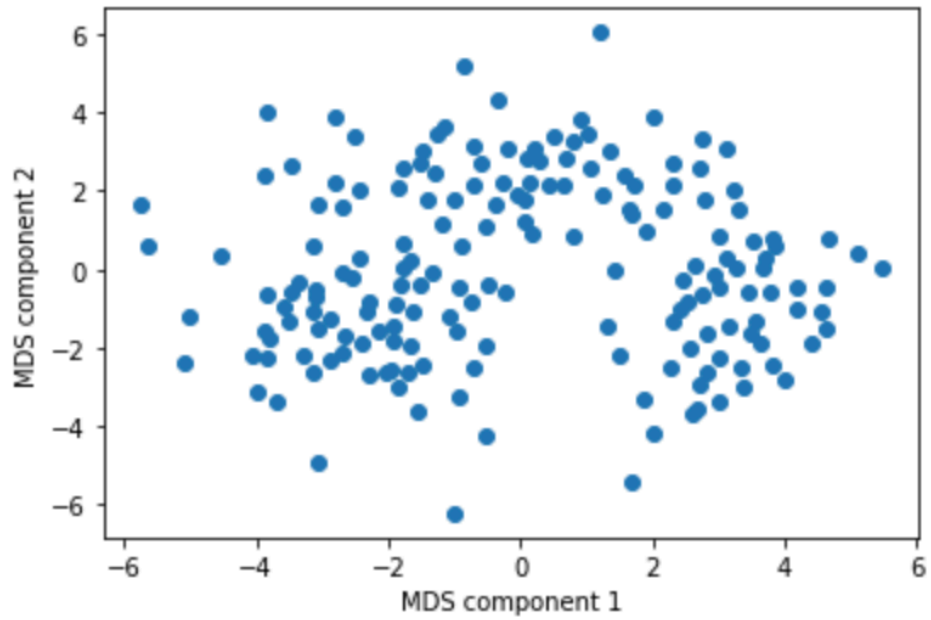
I use t-SNE for the data with n_components=2, fit_transform the normalized X in the data preprocessing process, plot the dimensionality reduced data in a 2D visualization. Running t-SNE with n_components = 2 will produce a 2D embedding on the data and the perplexity of 20 indicates that each point is considering its 20 nearest neighbors when constructing the probability distribution. Below is the plot between the 2 components:



It is indicated from the plot that there seems to be 3 different clusters of wine in the dataset.

Question 3

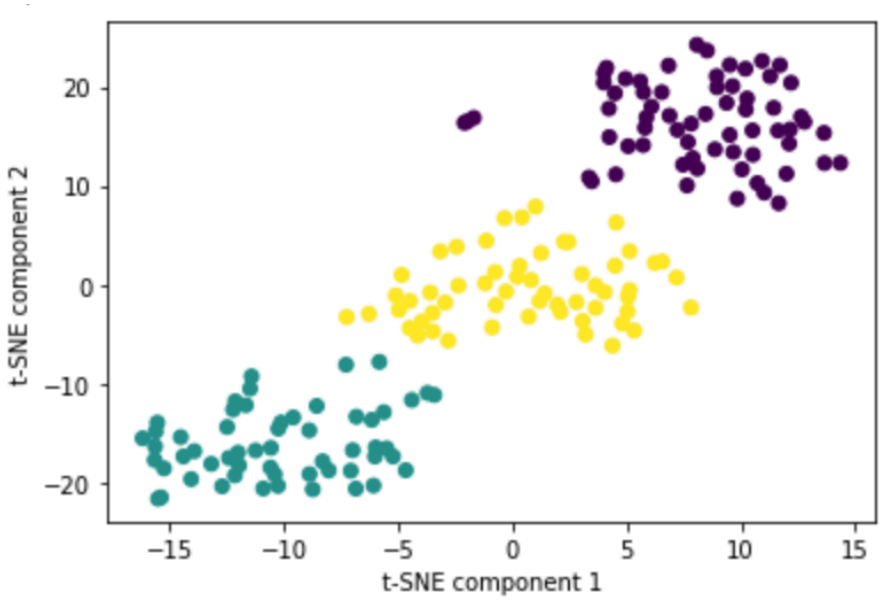
I use MDS for the data with n_components=2, fit_transform the normalized X in the data preprocessing process, plot the dimensionality reduced data in a 2D visualization. Running MDS with n_components = 2 will produce a 2D embedding on the data. Then I determine the resulting stress of this embedding by mds.stress_ function. Below is the plot between the 2 components:



The resulting stress of the MDS model is 21775.05186. The stress is a measure of how well the distances are preserved in the lower-dimensional space where from this observation the stress value is quite high. Comparing the plot between the MDS model and the t-SNE model, t-SNE model show a clear separation between the 3 different cluster of wines that are not clearly defined in the MDS model. This maybe due to the reason that MDS's goal is to preserve the global structure of the data without taking into account of the local structure of the data making it not as effective as t-SNE in separating clusters and revealing patterns in the data.

Question 4

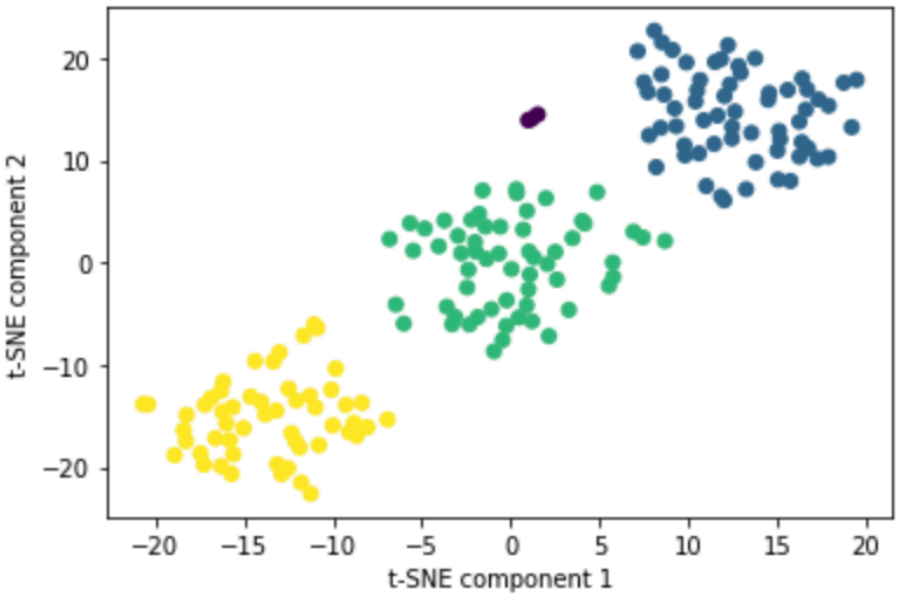
Building on the t-SNE dimensionality reduction model above with n_components = 2 and perplexity = 20, I use the silhouette method to determine the optimal number of cluster between the range 2 to 10 and then use kMeans with that determined number (k) to produce a plot that represents each wine as a dot in a 2D space in the color of its cluster and determine the total sum of the distance of all points to their respective clusters centers. The optimal K that is obtained from the kMeans is 3 indicating there exist 3 cluster of wines. Choosing the model based on the performance of the 3 models from Question 1 to Question 3, the t-SNE model has a clearer separation of the clusters that exist in the dataset. Therefore, the t-SNE model will be a better choice of model in this case. Below is the plot with color:



The total sum of the distance of all points to their respective clusters centers are 844.74359. This measures how well the clustering algorithm performed. The result obtained suggested that the t-SNE model performed fairly well.

Question 5

Building on the t-SNE dimensionality reduction model above with n_components = 2 and perplexity = 20, I use dbScan to produce a plot that represents each wine as a dot in a 2D space in the color of its cluster. I perform dbScan with epsilon of 4 and min_sample of 7. Then I fit_predict the embedding from t-SNE model with the initiated dbScan as labels to determine how many clusters will exist when I use dbScan. Epsilon is the maximum distance between 2 points for them to be considered part of the same neighborhood where eps = 4 indicates that the points within a radius of 4 units will be considered part of the same neighborhood whereas min_samples is the minimum number of points needed to form a dense region where min_samples = 7 indicates that a dense region must contain at least 7 points. The choice of eps and min_samples are appropriate based on the observed size of clusters observed in the t-SNE model plot. Below is the plot with color:



In the plot and in the number of clusters indicated from counting the number of unique labels after fit_predict, there seems to exist 4 clusters. However, the purple cluster is the outliers in the dataset that does not seems to match the other 3 dense clusters. This indicates that there are only 3 clusters of wines instead of 4.

Extra Credit 1

Given all consideration from the questions above, I am suggesting that there are 3 different kinds of wine in this dataset. In Question 1 (PCA model) and Question 3 (MDS model), there are a global structure of 3 different kinds of wines even thought there are not clear separation of the clusters whereas in Question 2 (t-SNE model), Question 4 (t-SNE with Silhouette and KMeans) and Question 5 (t-SNE with dbScan) there are a clear separation of 3 different kinds of wines where in Question 4 and Question 5 the clusters are even color-coded.

Extra Credit 2

There exist something in Question 1 that caught my attention, the PCA model gave me a insight of which features are more likely to be correlated to the first 2 principal components. This can be obtained by running pca.components_ function.

```
          0         1         2         3         4         5         6   \
0  0.144329 -0.245188 -0.002051 -0.239320  0.141992  0.394661  0.422934
1 -0.483652 -0.224931 -0.316069  0.010591 -0.299634 -0.065040  0.003360

          7         8         9         10        11        12
0 -0.298533  0.313429 -0.088617  0.296715  0.376167  0.286752
1 -0.028779 -0.039302 -0.529996  0.279235  0.164496 -0.364903
```

The correlation table above display the code that I ran in Question 1. This indicates that feature 7 (Flavonoids) is strongly and positively correlated with the first principal component whereas feature 10 (Color intensity) is strongly and negatively correlated with the second principal component. This implies that the feature “flavonoids” is the most important in explaining the variation in the data along the first principal component’s dimension whereas the feature “color intensity” is the most important in explaining the variation in the data along the second principal component’s dimension.