Algorithm 4. Expectation Maximization Clustering

Clustering involves the task of grouping data points into meaningful clusters, providing insights and aiding in various applications, including classification and anomaly detection. One powerful clustering technique is the Expectation-Maximization (EM) algorithm, often employed with Gaussian Mixture Models (GMM) for probabilistic clustering.

The EM algorithm is an iterative and probabilistic approach that iteratively refines cluster assignments based on a probabilistic model. It is particularly useful when dealing with data that can be represented as a mixture of underlying distributions. The core idea behind EM is to estimate the parameters of these underlying distributions iteratively. It consists of two main steps:

1. Expectation (E-Step):

In this step, the algorithm estimates the probability of each data point belonging to each cluster. It calculates the likelihood of data points given the current model parameters and assigns probabilistic cluster memberships.

2. Maximization (M-Step):

The M-Step updates the model parameters based on the expected cluster assignments from the E-Step. It aims to find the parameters that maximize the likelihood of the data under the current probabilistic model.

Gaussian Mixture Models:

Gaussian Mixture Models (GMM) are a versatile statistical tool for modeling complex data distributions. They are especially valuable in wine classification, where data can exhibit multiple underlying patterns. GMMs represent a blend of Gaussian distributions, each defining a unique cluster. Parameters like means, variances, and weights characterize these Gaussian components, capturing central tendencies, spread, and cluster importance. Within the implemented Expectation-Maximization (EM) algorithm, GMMs facilitate probabilistic clustering. The EM algorithm iteratively assigns wine samples to clusters, refines cluster parameters, and calculates classification probabilities, making it an effective approach for wine classification using a mixture of Gaussian distributions.

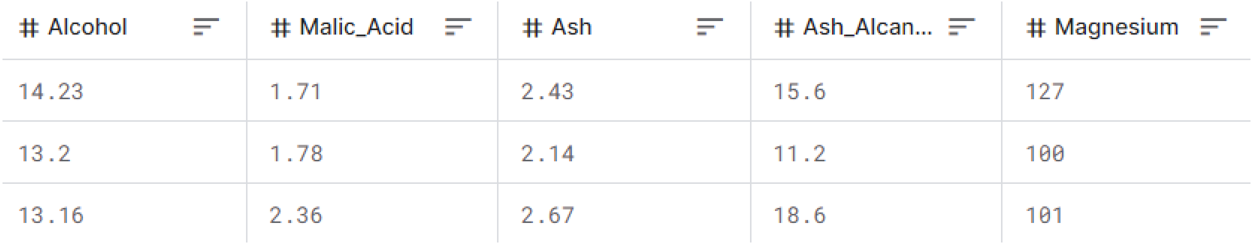
Dataset used:

The dataset used by us is adapted from https://archive.ics.uci.edu/ml/datasets/wine. It is a dataset of three different types of wines - Red, White and Rose. It has 13 attributes, which are all of integer or float data types. The following descriptions are adapted from the UCI webpage:

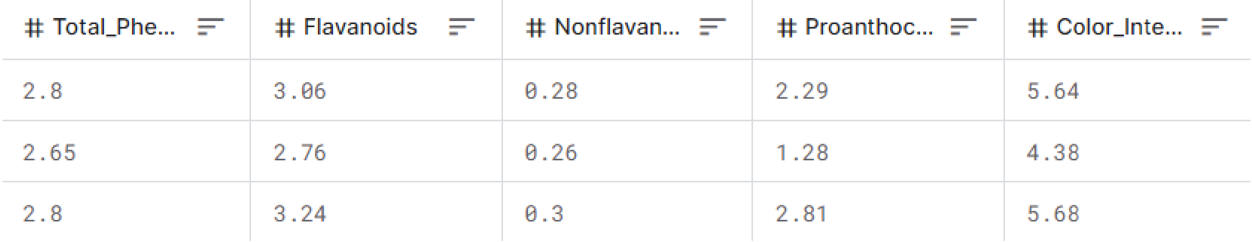
These data are the results of a chemical analysis of wines grown in the same region in Italy. The analysis determined the quantities of 13 constituents found in each of the three types of wines. The attributes are:

Alcohol content, Malic acid content, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline profile.

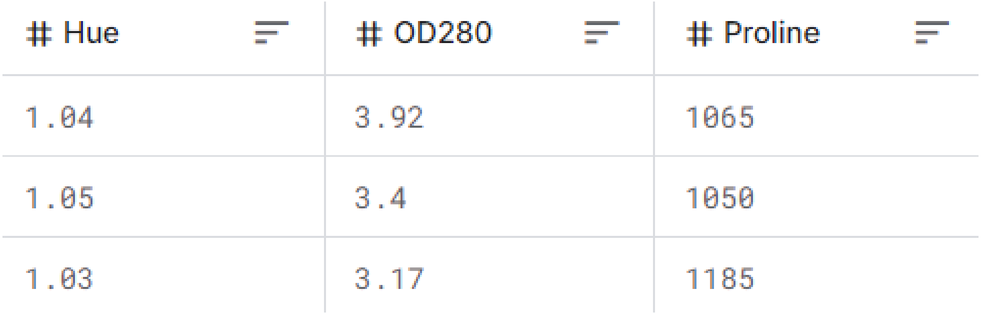
Snapshot of the dataset(all 13 attributes of the first 3 entries) :



# Alcohol	# Malic_Acid	# Ash	# Ash_Alcan...	# Magnesium
14.23	1.71	2.43	15.6	127
13.2	1.78	2.14	11.2	100
13.16	2.36	2.67	18.6	101



# Total_Phe...	# Flavanoids	# Nonflavan...	# Proanthoc...	# Color_Inte...
2.8	3.06	0.28	2.29	5.64
2.65	2.76	0.26	1.28	4.38
2.8	3.24	0.3	2.81	5.68

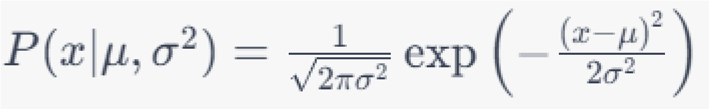


# Hue	# OD280	# Proline
1.04	3.92	1065
1.05	3.4	1050
1.03	3.17	1185

Working of the model:

The Expectation-Maximization (EM) algorithm, augmented with Gaussian Mixture Models (GMM), operates as follows. In the Initialization Step (I-Step), wine samples are initially assigned to clusters, providing a starting point for the algorithm. The core of the algorithm comprises two main steps - the Expectation Step (E-Step) and Maximization Step (M-Step).

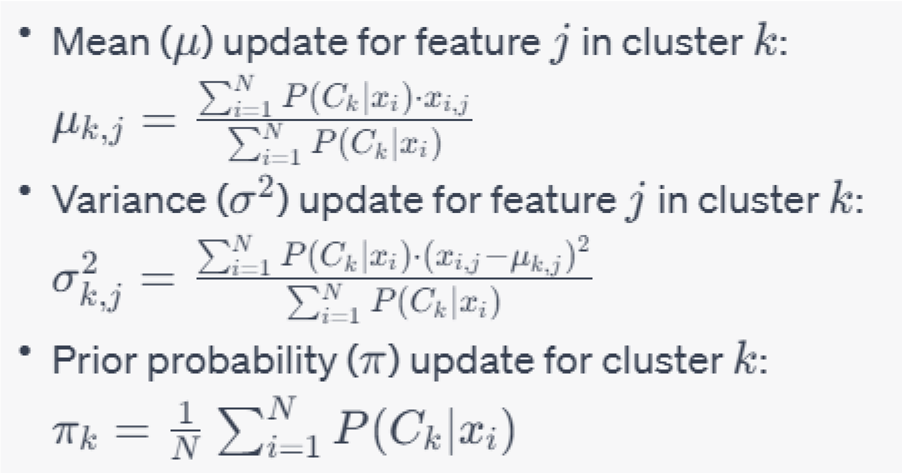
1. E-Step (Expectation Step): In the E-Step, the algorithm calculates the probability of each wine sample belonging to each cluster. It computes these probabilities based on the current model parameters and assigns probabilistic cluster memberships. Using the Gaussian probability density function (PDF) formula for a single feature:



$$P(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

The algorithm computes the likelihood of a wine sample for each cluster and updates the probabilities of cluster assignments for all samples.

2. M-Step (Maximization Step): The M-Step refines the model parameters to better fit the data. It updates the means, variances, and prior probabilities for each cluster based on the expected cluster assignments from the E-Step. The formulas for updating these parameters are as follows:



- Mean (μ) update for feature j in cluster k:

  $$\mu_{k,j} = \frac{\sum_{i=1}^{N} P(C_k|x_i) \cdot x_{i,j}}{\sum_{i=1}^{N} P(C_k|x_i)}$$

- Variance (σ^2) update for feature j in cluster k:

  $$\sigma_{k,j}^2 = \frac{\sum_{i=1}^{N} P(C_k|x_i) \cdot (x_{i,j} - \mu_{k,j})^2}{\sum_{i=1}^{N} P(C_k|x_i)}$$

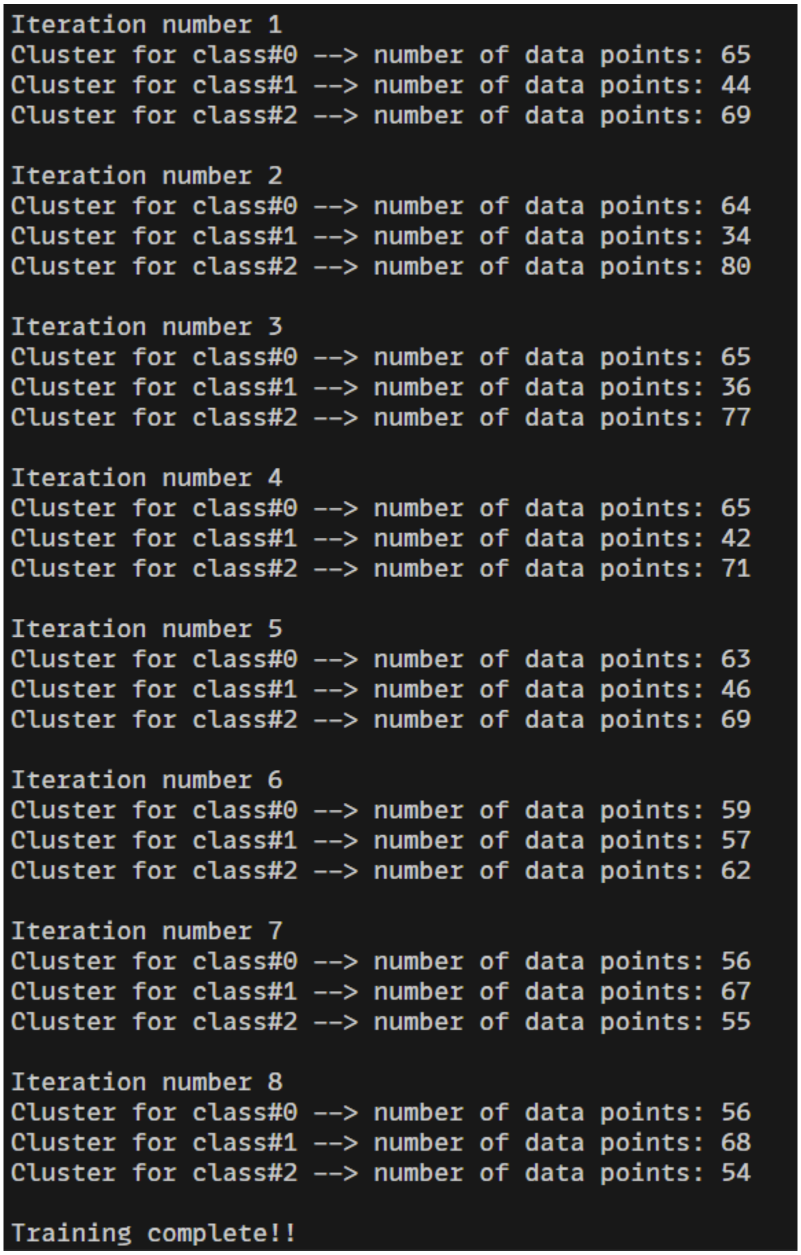
- Prior probability (π) update for cluster k:

  $$\pi_k = \frac{1}{N} \sum_{i=1}^{N} P(C_k|x_i)$$

3. Termination Condition: The algorithm iteratively cycles between the E-Step and M-Step until convergence. The termination condition is often based on a maximum number of iterations, a threshold on parameter updates, or when cluster assignments no longer change significantly between iterations. Once this condition is met, the algorithm terminates, providing a refined clustering solution for the wine samples.

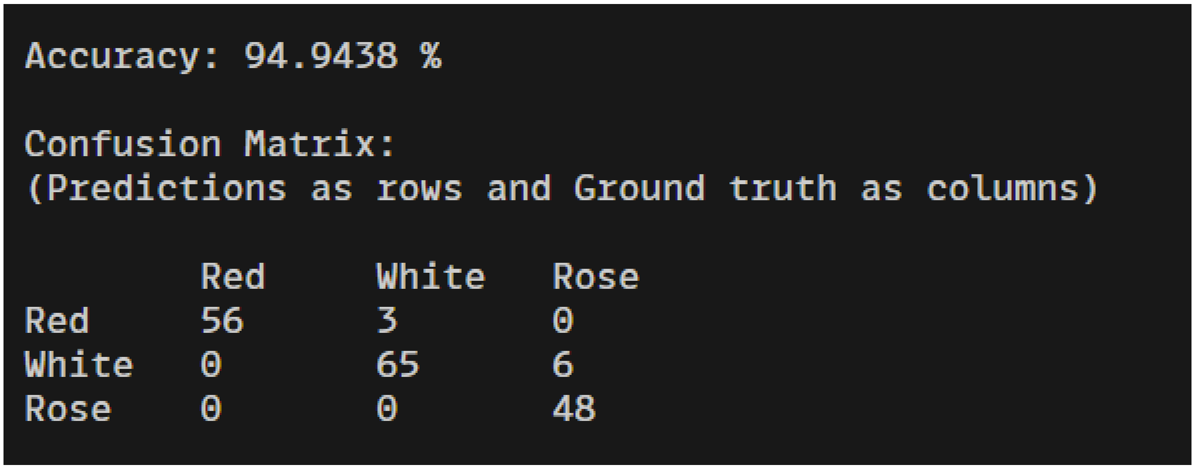
Final Results:

Upon running our algorithm on the given dataset, we observed that the algorithm completes in 8 iterations. This means that after the 8th iteration, the new calculations for mean and variance did not affect the assigned cluster for any of the training examples. As there is no change, the algorithm terminates. The clustering details for each iteration can be seen as follows:-



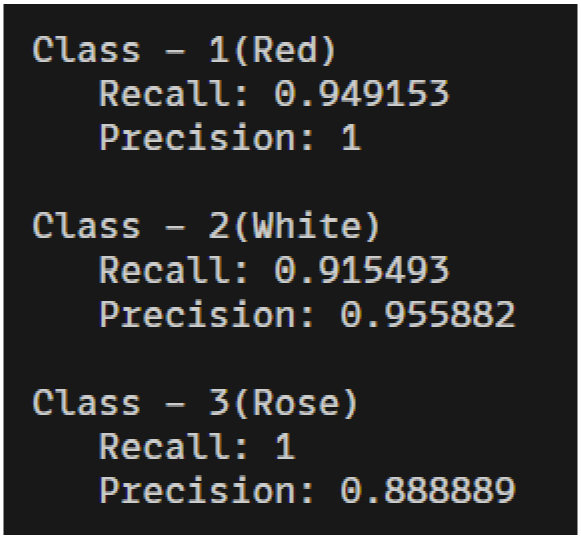
```
Iteration number 1
Cluster for class#0 --> number of data points: 65
Cluster for class#1 --> number of data points: 44
Cluster for class#2 --> number of data points: 69

Iteration number 2
Cluster for class#0 --> number of data points: 64
Cluster for class#1 --> number of data points: 34
Cluster for class#2 --> number of data points: 80

Iteration number 3
Cluster for class#0 --> number of data points: 65
Cluster for class#1 --> number of data points: 36
Cluster for class#2 --> number of data points: 77

Iteration number 4
Cluster for class#0 --> number of data points: 65
Cluster for class#1 --> number of data points: 42
Cluster for class#2 --> number of data points: 71

Iteration number 5
Cluster for class#0 --> number of data points: 63
Cluster for class#1 --> number of data points: 46
Cluster for class#2 --> number of data points: 69

Iteration number 6
Cluster for class#0 --> number of data points: 59
Cluster for class#1 --> number of data points: 57
Cluster for class#2 --> number of data points: 62

Iteration number 7
Cluster for class#0 --> number of data points: 56
Cluster for class#1 --> number of data points: 67
Cluster for class#2 --> number of data points: 55

Iteration number 8
Cluster for class#0 --> number of data points: 56
Cluster for class#1 --> number of data points: 68
Cluster for class#2 --> number of data points: 54

Training complete!!
```

Upon comparing our predictions with the ground truth labels, we arrived at the following conclusions :-



```
Accuracy: 94.9438 %

Confusion Matrix:
(Predictions as rows and Ground truth as columns)

        Red     White   Rose
Red     56      3       0
White   0       65      6
Rose    0       0       48
```

Based on our confusion matrix, here are the precision and recall values for each of the 3 classes :-



```
Class - 1(Red)
   Recall: 0.949153
   Precision: 1

Class - 2(White)
   Recall: 0.915493
   Precision: 0.955882

Class - 3(Rose)
   Recall: 1
   Precision: 0.888889
```