Name: Tuguldur Ts. 鐵特德 **Professor**:孫民

Student ID:109006271 **Content**: Homework 1 Programming Report

Course: Machine Learning

Homework 1 Programming Report

Part 2.1)

To split the "wine.csv" into "train.csv" and "test.csv", I used a method to randomize the three wine types (0,1,2) within their respective types. From each shuffled group, the first 20 samples are allocated to the test dataset, and the remaining samples are used for training. After testing and making sure that we have 60 for testing and 423 for training, I save the datasets to their respective csv.

Part 2.2)

Predictions are made for each sample in X_test by applying the calculate_posterior function to each row. The class with the highest posterior probability is chosen as the prediction. The accuracy of the predictions is calculated by comparing the predicted classes (predictions) with the actual classes (y_test).

To solve this problem first I calculate the **prior**, **mean** and **variance** for each feature within each class, **Gaussian Likelihood,** and then finally **posterior**. Where from the references I found online, I calculate the Posterior Probability as Prior(Y=C) +Likelihood(X|Y=C).

For Prior Calculation, I follow the formula: $P(Y = C) \Rightarrow \frac{\#Num\ of\ samples\ with\ class\ C}{Total\ sample\ \#Num}$

Where our **total sample #Num** is 483 **(Table 2.2.1 Calculation Table)**.

Type	Python Calculation
0	0.366430
1	0.437352
2	0.196217

(Table 2.2.1 Calculation Table)

Where the calculation as follows:

$P(Y = 0) \Rightarrow \frac{155}{423}$

$P(Y = 1) \Rightarrow \frac{185}{423}$

$P(Y = 2) \Rightarrow \frac{83}{423}$

For mean and variance values, I use the formula:

$$\mu_{c,j} = \frac{1}{n_c}\sum_{i=1}^{n_c} x_{i,j}$$

$$\delta^2_{c,j} = \frac{1}{n_c}\sum_{i=1}^{n_c} (x_{i,j} - \mu_{c,j})^2$$

By calculating the mean and variance, we can easily calculate the Gaussian Likelihood where the formula that I used is:

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

(Snippet of the Gaussian Likelihood Function shown in **Figure 2.2.2 Accuracy Figure**).

Then, the Posterior Probability was calculated to find the accuracy where it was 96.67% (**Figure 2.2.2 Accuracy Figure**). The method to find the accuracy was mentioned in above. The full code with the comments can be found in my HW1.py file.

```
    # Testing for question 4
    #accuracy_modified = (predictions_modified == y_test).mean()

    # Output the results
    print(f'Original Accuracy: {accuracy_original:.2%}')
    # Testing for question 4
    #print(f'Modified Accuracy: {accuracy_modified:.2%}')
15]  ✓ 0.0s

··  Original Accuracy: 96.67%
```

(Table 2.2.2 Accuracy Figure)

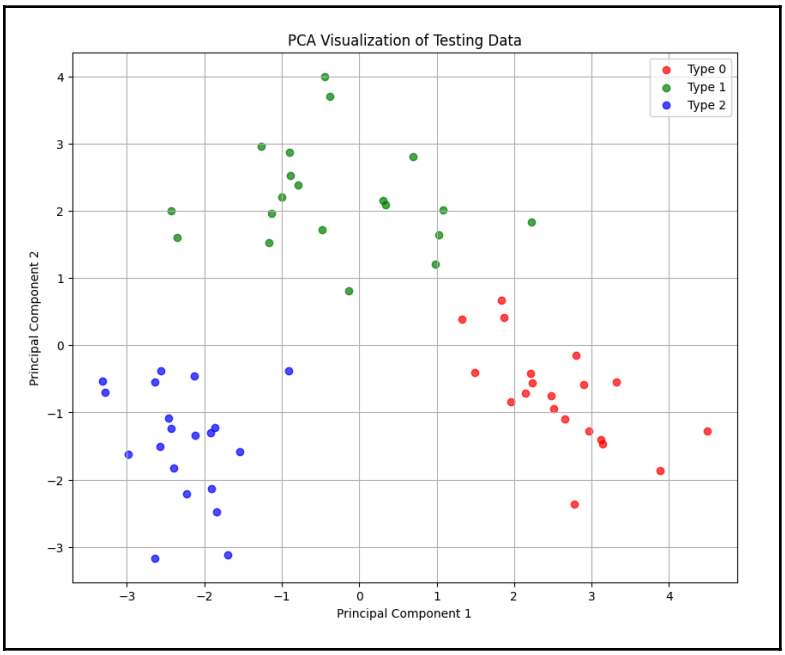
Part 2.3)

The visuals for the PCA in 2D and 3D can be found below in **Figure 2.3.1** and **Figure 2.3.2** respectively.

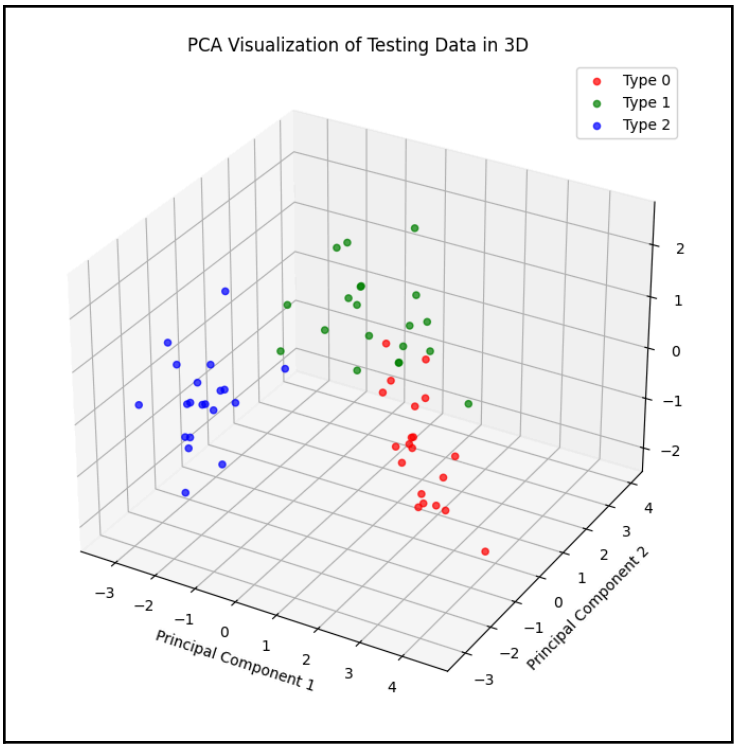
PCA is widely used to reduce dimensionality in a dataset especially in high dimensional datasets. Then beginning with standardizing the data and through eigen decomposition of the covariance matrix the PCA principle components are defined and it eliminates unrelevant features and simplifies and enhances the complex data by capturing the features that explain variation. This process aligns the data to a new coordinate system in which the first few axes of the data contains the major portion of variance and the structure of the data can be used to simplify the analysis and visualization.

The PCA diagrams appear to have a good degree of success of separating the data into categories (Red, Green, Blue) with whatever axis is being demonstrated; there is a notable amount of separation in the 2D diagram between the red and blue points. This shows that PCA has captured the variance that differentiates among such groups as hypothesized earlier. Some green and red points are positioned closely to each other, which may indicate objects that are less distinguishable in terms of the chosen categories; still, the 3D view of the data provides for clear separation of categories with the help of the third dimension. This way of spatial disposition permits the multiple viewpoints concerning relations and clustering and reveals the role of PCA

in the dimensionality decrease while retaining important variations for the subsequent operations, including clustering.



(Figure 2.3.1 2D PCA visuals)



(Figure 2.3.2 3D PCA visuals)

Part 2.4)

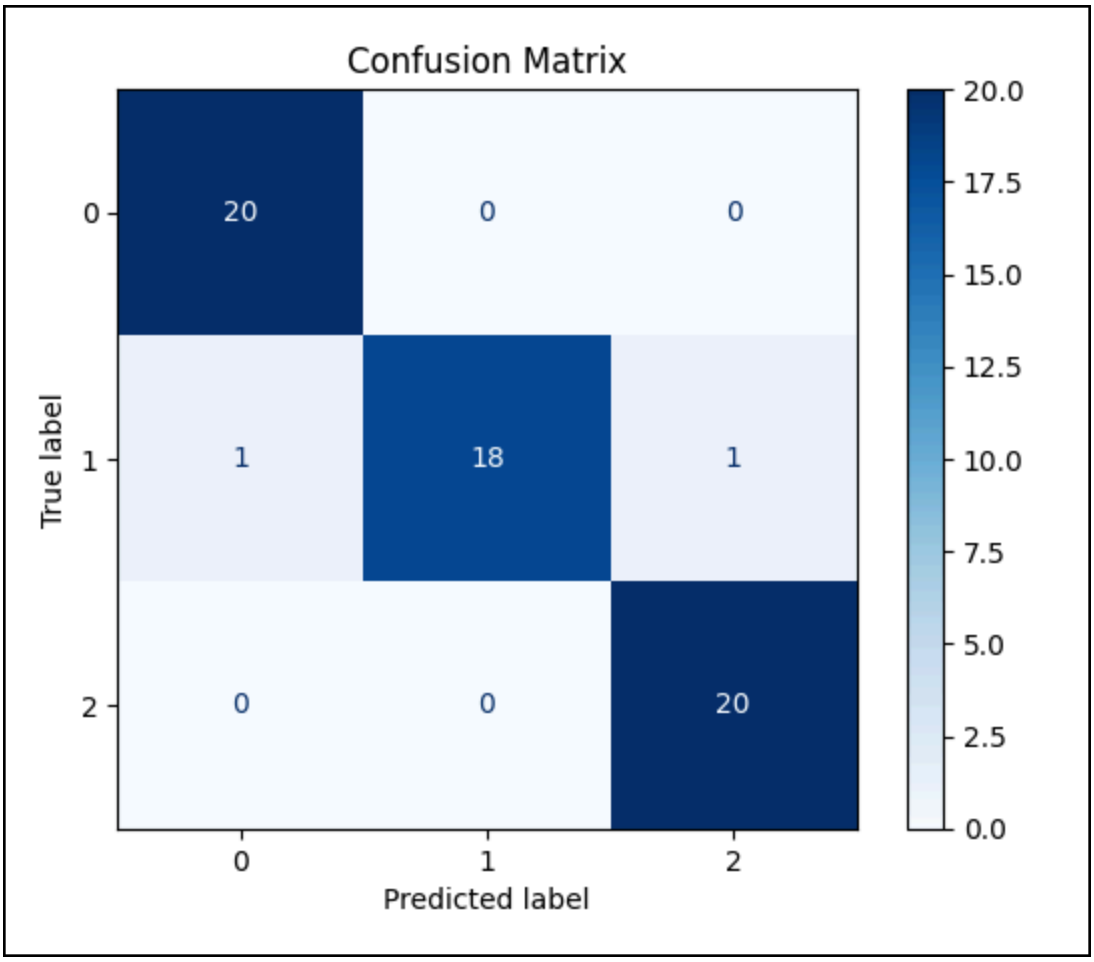
The output indicates that the prior distribution affects the values of the posterior probabilities in the Bayesian classification model. As can be seen, there is only a marginal change in classification accuracy when prior distribution is changed from data derived values to, uniform values to be precise 1/3 for each class. This observation can be used to show that while the prior distribution is useful in establishing the starting point of the calculation, its impact at the end rarely compares to the likelihood derived from the relevant data. This means that the prior does not dominate in cases where there is a likelihood especially when the class discrimination is well defined. The result of the modified code for part 2 and my proof of reasoning can be seen in the **Figure 2.4.1** below.

```
Class 2: Prior log probability = -1.0986, Sum of log likelihood = -17.0239, Posterior log probability = -18.1225
Evaluating instance with features: [1.2790e+01 2.8900e+00 2.5900e+00 2.3790e+01 1.0008e+02 2.2800e+00
 5.4000e-01 5.2000e-01 7.7000e-01 4.7700e+00 8.8000e-01 2.1000e+00
 5.6982e+02]
Class 0: Prior log probability = -1.0986, Sum of log likelihood = -37.9720, Posterior log probability = -39.0706
Class 1: Prior log probability = -1.0986, Sum of log likelihood = -26.1084, Posterior log probability = -27.2070
Class 2: Prior log probability = -1.0986, Sum of log likelihood = -20.5141, Posterior log probability = -21.6127
Evaluating instance with features: [1.323e+01 2.100e+00 2.570e+00 2.306e+01 9.885e+01 1.280e+00 4.100e-01
 3.400e-01 1.290e+00 7.660e+00 6.400e-01 1.490e+00 6.145e+02]
Class 0: Prior log probability = -1.0986, Sum of log likelihood = -42.6878, Posterior log probability = -43.7864
Class 1: Prior log probability = -1.0986, Sum of log likelihood = -37.6533, Posterior log probability = -38.7519
Class 2: Prior log probability = -1.0986, Sum of log likelihood = -14.0075, Posterior log probability = -15.1061
Evaluating instance with features: [1.3180e+01 4.2900e+00 2.7800e+00 2.4080e+01 1.0805e+02 1.8100e+00
 6.8000e-01 2.4000e-01 9.5000e-01 7.8600e+00 6.1000e-01 1.7200e+00
 5.4301e+02]
Class 0: Prior log probability = -1.0986, Sum of log likelihood = -45.2220, Posterior log probability = -46.3206
Class 1: Prior log probability = -1.0986, Sum of log likelihood = -42.0090, Posterior log probability = -43.1076
Class 2: Prior log probability = -1.0986, Sum of log likelihood = -17.9884, Posterior log probability = -19.0870
Original Accuracy: 96.67%
Modified Accuracy: 96.67%
```

(Figure 2.4.1 Modified Part 2)

Part 2.5)

The confusion matrix indicates that the classifier has performed exceptionally well in distinguishing between the three classes, achieving high accuracy with 58 out of 60 predictions being correct. Class 1 had one instance misclassified as Class 0 and another as Class 2, but there were no misclassifications for Classes 0 and 2. Overall, the results suggest that the classifier is highly effective with a very low error rate. My result can be seen in the **Figure 2.5.1** Below.



(Figure 2.5.1 Confusion Matrix Result)

References:

https://www.geeksforgeeks.org/principal-component-analysis-pca/

https://www.youtube.com/watch?v=IW4T7p4ZPxQ

https://stackoverflow.com/questions/55509041/loglikelihood-of-normal-distribution

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html