Matric number: s1632798

Task 3

Task 3.2

The elapsed time for my_gaussian_classify () is around 10.516264seconds.

The requested information:

Number of test samples	Number of wrongly classified test samples	Accuracy
7800	1232	0.8421

Task 3.3

This task is about increasing the accuracy of the Gaussian distribution. There are many ways that we can improve the accuracy with. However, without knowing what these methods are supposed to do and how we can use them effectively we may not increase our accuracy and even more we can greatly decrease it. So before adopting different techniques for our algorithm, we have to do research so we can maximize our improvement. One of these techniques is the one we used in Task 3.3 for modifying the classifier developed in Task3.1 - the dimensionality reduction with PCA. Principal Component Analysis (PCA) uses the variance of each feature to maximize its separability. In this context, PCA is an unsupervised algorithm. The idea behind PCA is simply to find a low-dimension set of axes that summarize data. PCA does not take information of classes into account, it just looks at the variance of each feature because it assumes that features that present high variance are more likely to have a good split between classes. Often, people end up making a mistake thinking that PCA selects some features out of the dataset and discards others. The algorithm actually constructs new set of properties based on combination of the old ones. Mathematically speaking, PCA performs a linear transformation moving the original set of features to a new space composed by principal components. The algorithm uses the concepts of a variance matrix, a covariance matrix, eigenvectors and eigenvalues pairs to perform PCA, providing a set of eigenvectors and its respectively eigenvalues as a result. The eigenvectors represent the new set of axes of the principal component space and the eigenvalues carry the information of quantity of variance that each eigenvector has. So, in order to reduce the dimension of the dataset we are going to choose those eigenvectors that have more variance and discard those with less variance. So after knowing what PCA actually is and how it works we have to ask ourselves if it is really going to improve our accuracy and how. It is often the case that a small subset of the dimensions actually accounts for most of the variance, meaning we can reduce our data to those dimensions and still understand the patterns presented well - in some cases this may even make the patterns more obvious than the original dataspace, which is why PCA dimensionality reduction may improve classification. Of course, we will lose some of the information when we reduce the dimensions. There is some intelligent choice needed in choosing how many PCA dimensions to keep. This is done by investigating, researching and thinking on how our approach may impact the accuracy on unseen data. So if we choose the right amount of dimensions to keep, the information we will lose is unnecessary and it is called “noise”. We can use PCA for denoising. We will get data with less features that will contain the most relevant information.

PCA makes patterns more obvious, aiding pattern recognition. We have to apply PCA on the training data in order to learn the pattern. And maybe the most difficult and trickier question here is what we should do with the test data. Should we combine the training and the test data? Should we apply PCA to the training data separately? Or should we use the principal components from the training data? Without research these questions can become really hard. Combining the data is clearly incorrect, because if we run PCA on the two sets separately, we will end up with two different spaces. We cannot train a classifier in one space, and apply it to a different space. At first look we can think that doing PCA separately on the training data and the test data is a good idea. But this is “cheating”. When we train a classifier, we cannot use any information from the test set. So the correct way would be to run PCA on the training set, save the principal components that we use, and then use them to transform the points in our test set. This way the points in both sets end up in the same space, and we are not using any knowledge about our test set during training. Alternatively, we can use an entirely separate data set, just for computing the principal components. Then we will project both our training set and our test set into the space defined by those. So PCA can be a powerful tool for decreasing the number of features and therefore the complexity of our data, increasing the accuracy of the classifier and reducing the noise. So after we did the PCA the right way and made the right choice on how many PCA dimensions to keep (through trial and error) we managed to increase the accuracy of the classifier with 3% - from 84% to 87%.

Dimensionality Reduction plays a really important role in machine learning, especially when we are working with thousands of features. Principal Components Analysis is one of the top dimensionality reduction algorithm, it is not hard to understand and use it in real projects. This technique, in addition to making the work of feature manipulation easier, it still helps to improve the results of the classifier, as we saw in the task.

We can adopt even more techniques and make more improvements to our classifier. For example, we can use clustering with Gaussian Mixture Models. Gaussian Mixture Models are a clustering technique that allows us to fit multivariate Gaussian distributions to our data. That way using the k-means clustering algorithm we can obtain multiple Gaussian distributions per class. The idea behind Gaussian Mixture Models is to find the parameters of the Gaussians that best explain our data. This is what we call generative modeling. We are assuming that these data are Gaussian and we want to find parameters that maximize the likelihood of observing these data. In other words, we regard each point as being generated by a mixture of Gaussians and can compute that probability. There are a lot of other techniques that we can adopt. However, not all of them will work well for our data. The hard part here is to research, find the proper ones and add them to our classifier so they can not interfere, but benefit from each other.