Inf2B – CW2
S1210313

4. Report:

3.1 – K Nearest Neighbors

Confusion Matrix:

X confused for y	1	2	3	4	5	6	7	8	9	10
1	0	0	10	4	2	1	2	2	1	4
2	0	0	0	0	0	6	0	0	0	4
3	0	1	0	1	1	1	2	3	5	13
4	2	0	7	0	3	0	0	5	7	15
5	4	0	1	5	0	0	2	16	8	3
6	5	9	3	1	0	0	1	0	1	3
7	1	0	2	2	1	3	0	3	0	15
8	0	0	5	8	10	0	2	0	5	6
9	2	0	9	5	13	0	1	7	0	6
10	0	1	16	11	3	1	4	1	3	0

- ***Are all genres equally difficult to classify?***

No, some genres are more challenging to classify than others. This makes sense, as some of the features in one genre may be similar to another if there are common practices in audio-engineering, or recording standards.

- ***What is the most common error?***

Classifying 10 for 3, or rock music for country music

- ***Classification accuracy***

310 items were labeled incorrectly out of a total of 929 items, achieving an accuracy of approximately 67% for K = 1.

- ***Complexity for training and classification***

Naive k nearest neighbor classification can be implemented in O(nd) time, where n is the cardinality of the training set and d is the dimensionality of the size of the space. The training could be considered the calculation of the k nearest neighbors and accounts for most of the complexity of the classification.

- ***Proposal for further expansions:***

For small data sets with few features, this is an acceptable method of classification, however, as the size of the feature-space increases, so does the computational complexity. If the data was sorted in a meaningful way it would be possible to find the nearest neighbors in less time. Also, there may be better heuristics than distance for classification which could be used.

Inf2B – CW2
S1210313

3.2

A) **What is the prior probability for each class? What classification accuracy would we obtain if we classified each music fragment by choosing a label at random?**

Since there is a uniform distribution the probability is *1/n*, where *n* is the number of classes. Assuming uniform prior distribution over all classes there is a probability of *P(C) = 1/10*. Given this there is a probability of *1/10* of choosing the correct classification of a song by picking a random classification. Using this, we can eliminate the prior probability from the probability calculation, as it will be an added constant for calculated probability value, regardless of class.

B) **Diagonal Covariance Gaussian Model :**

- ***What is the classification accuracy?***

My implementation: 423 incorrect of 929 cases, or an accuracy of approximately 54%

- ***Are the classification boundaries of this method linear?***

No, the classification boundaries will not be linear. Given the covariances are diagonal, the Gaussians will have 124 dimensional oval-like shapes. Thus, the boundaries between classes cannot be linear.

C) **Gaussian Model With Equal Full Covariance:**

- ***What is the classification accuracy?***

My implementation: 277 incorrect of 929 cases, or an accuracy of approximately 70%

- ***What is the shape of the classification?***

Given each class shares an equal covariance, there is a collection of 10 Gaussians—the number of classes—which are 124 dimensional—the number of used features. Each is centred around the mean vector of the. Thus, if we were to visualize the classification over all the possible values, it would be an 124 dimensional space where the decision boundaries are where the Gaussians meet and have equal value. So, the spaces on either side of these boundaries are the.

- ***Complexity of training and classification for Gaussians***

For training, the training set must be separated into its classes—in its most efficient form this can be done in O(n) where n is the length of the training set— and then using this, the average of each class is calculated, which involves iterating over the whole of the matrix, or length by width. This can be done in O(mn) where n is the length of a class matrix and m is the width. Following this, the covariance matrix is calculated. As described, there is a high complexity for training.

For classification, a function is used to calculate the probability for each class, the largest of which is the result, which can be done in O(n), where n is the number of classes. This demonstrates that while the training is intensive, once the values are calculated, the actual classification is relatively efficient.

- ***Proposal for further expansions:***

For any of these methods, it would be beyond the scope of the coursework to have to deal with any situation where the data is incomplete. Even so, in a production system, especially one dealing with older data, or data which may have been poorly maintained and collected, it’s likely that older labels may be missing, or inaccurate. Further experimentation could identify and test ways of coping with partially labelled and poorly label data.

- ***Substantiated recommendation for a method to use in a production system***

K NN means would require that the nearest neighbors be calculated every time the process is run—something which is clearly computationally intensive in most practical settings; data sets will be too large to be efficient.

With the Gaussian models, the computational overhead is largely the same. One of their greatest features is that while it may be intensive to calculate the mean, or the covariance matrix of a particular training set, so long as the training set is not changing, it is only necessary to calculate these values once.

Having calculated the mean and the covariance values for either of the Gaussian implementations, it would be possible store the values and then use them to classify any new data. However, if the training set were to be added to, to improve the accuracy with this new information a new covariance and mean values would have to be calculates..

Given the lower computational over-head—after training—I would recommend a Gaussian model to classify new information, specifically a Gaussian model with shared covariance.