S1443483

INF2B-CW2 REPORT 2

Task 2

3.1

I wrote the code for k-nearest neighbours classification. For k =1 neighbour we have:

Confusion Matrix =

85	0	0	4	1	1	6	0	2	1
2	84	4	2	0	1	3	1	2	1
1	1	85	1	5	1	1	1	3	1
2	1	1	91	1	1	1	1	0	1
0	2	3	0	87	3	0	4	1	0
0	0	3	0	1	84	0	4	1	7
8	2	2	0	0	1	86	1	0	0
0	0	4	2	3	7	0	80	2	2
0	3	2	2	0	2	1	6	81	3
3	1	1	1	0	2	0	0	3	89

Correct Classification rate = 0.8520

3.2.1

The determinant of the covariance matrix of each class is:

Determinant for class 1 = 2.8386e-235

Determinant for class 2 = 4.6100e-262

Determinant for class 3 = 2.5598e-237

Determinant for class 4 = 4.3944e-239

Determinant for class 5 = 3.2206e-260

Determinant for class 6 = 2.7535e-243

Determinant for class 7 = 1.4430e-249

Determinant for class 8 = 1.8820e-241

Determinant for class 9 = 1.2662e-242

Determinant for class 10 = 2.5282e-253

Confusion Matrix =

```
90   0   1   5   0   0   3   0   0   1
 1  86   2   2   0   1   4   1   3   0
 1   3  79   2   3   0   6   2   4   0
 1   1   0  92   3   0   2   0   0   1
 0   0   2   1  87   4   0   2   3   1
 1   0   5   1   0  87   1   3   1   1
 2   1   0   1   0   0  95   1   0   0
 1   0   4   1   1  10   1  79   1   2
 0   2   3   2   1   0   2   1  85   4
 4   1   1   0   1   2   1   0   2  88
```

Correct Classification Rate = 0.8680

3.2.2

Determinant = 7.4070e-173

Confusion Matrix =

```
83   0   1   4   2   1   7   0   1   1
 0  81   6   3   0   1   2   3   3   1
 0   2  91   0   0   0   1   5   1   0
 3   0   5  85   2   0   1   1   1   2
 0   0   9   0  77   4   0   8   2   0
 1   0   3   1   0  87   0   6   1   1
 1   4   2   1   0   1  89   2   0   0
 0   0   6   0   0   1   1  89   2   1
 0   2   4   1   1   2   0   2  85   3
 2   0   2   0   0   3   0   2   2  89
```

Correct Classification Rate = 0.8560

3.3

I conclude from my experiments that the Gaussian Models classification using Maximum Likelihood Estimation have a better correct classification rate than the k-nearest neighbours classification. For the Gaussian Models, taking a full covariance matrix for each class (each class has a different covariance matrix) results in a more accurate classification rate than taking a shared covariance matrix for all classes (all classes share the same covariance matrix). The difference between knn classification and classification using Gaussian models is following: In knn, we calculate the distance between the test data and the training data and we classify the test data depending on the class of the k points nearest to the test data points. In Gaussian Models however, we use the Maximum Likelihood Estimation to calculate the probabilities between the test data and training data. In the Gaussian Model with separate covariance matrix for each class, I calculated the log probabilities using the suitable function. In the Gaussian Model with shared covariance for all classes, I used the linear discriminant function.

An advantage of the k-nearest neighbours classification is that complex concepts can be learned by local approximation using simple procedures. I achieved a fairly good correct classification rate by just using the simple knn algorithm. A disadvantage is that the performance depends on the number of dimensions that we have. Also, it takes up a lot of memory to run if we have many instances (I noticed that my laptop froze at times when running the knn algorithm for many test feature vectors). Gaussian Model classifiers give a predictive variance estimate around our prediction and they have a clear probabilistic interpretation, thus giving us better correct classification rate opposed to knn (as I noticed from my own classification results).

In the k-nearest classification, one modification we can do to reduce the classification errors is take a bigger k.