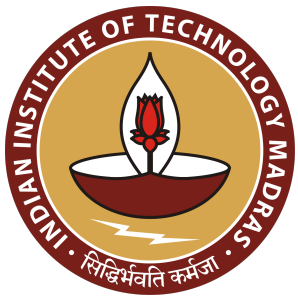
ASSIGNMENT 2

CS5691 Pattern Recognition and Machine Learning

CS5691 Assignment 2

Team Members:	
BE17B007	N Sowmya Manojna
PH17B010	Thakkar Riya Anandbhai
PH17B011	Chaithanya Krishna Moorthy

Indian Institute of Technology, Madras



Contents

1 Dataset 1A

1.1 K-nearest Neighbors Classifier

1.1.1 Mathematical Formulation

The K Nearest Neighbor is a statistically non-parametric model that can be used for regression as well as for classification. It assumes that similar things exist in close proximity. Crucial steps in a K-Nearest Neighbor classifier are:

- A distance metric is first specified, the most commonly used metric is the euclidean distance:

$$d = ||\vec{x_1} - \vec{x_2}|| \tag{1}$$

  where $||.||$ denotes the norm function. Other commonly used distance metrics are the Manhattan distance and Cosine similarity. For our application, Euclidean distance is used.
- Using the specified distance metric, the distance between the test instance and each training example is evaluated.
- The class label that occurs most frequently amongst the nearest k training examples is assigned to the test instance.

Advantages of KNN are:

- KNN does not require a training period, it just stores the training dataset and learns from it at the time of making a prediction, hence it is generally much faster than other classification algorithm.
- Since the algorithm does not require prior training, new data points can be added seamlessly.
- Easy to implement,the number of parameters are just two: k and the distance metric to be used.

Disadvantages of KNN are:

- Computationally expensive for large datasets or high number of features, since the distance is evaluated between test point and all the points in the training dataset.
- Sensitive to noisy data and outliers. Generally, increasing the value of k reduces the effect of noise.

1.1.2 Pre-Processing

The dataset 1A has 4 unique class labels - $[0.0, 1.0, 2.0, 3.0]$ as shown in Figure 2. Number of examples corresponding to each class label is 200. The train dataset is of dimension $(800, 3)$ while the CV dataset is of dimension $(120, 3)$. The third column in both datasets is the class label, while the first two columns are the real valued feature vectors - x_1 and x_2.

- There are no null values in the datasets.
- The rows of dev dataset are shuffled and further split into cross-validation and test data in the ratio of 70:30
- Range of x_1 is $(-11, 11)$ and range of x_2 is $(-12, 7)$. Since the ranges are almost similar, no feature scaling is required.

1.1.3 Model performance across k

The model was evaluated for k values: $[1, 7, 15]$. We find that irrespective of the value of hyperparameter k, the model obtained an accuracy of 100 over training data, cross-validation data as well as the test data.

Since model performance is best irrespective of k, the accuracy table, confusion matrix and decision boundary plot are all evaluated using $k = 1$ as to minimize the run time.

The accuracy table and confusion matrix are:

k-value	Train Accuracy	CV Accuracy	Test Accuracy
1	100	100	100
7	100	100	100
15	100	100	100

Table 1: Accuracy table for dataset 1A - KNN Classifier

Figure 1: Confusion matrix for $k = 1$, Train and Test dataset on left and right respectively

1.1.4 Decision Region Plot

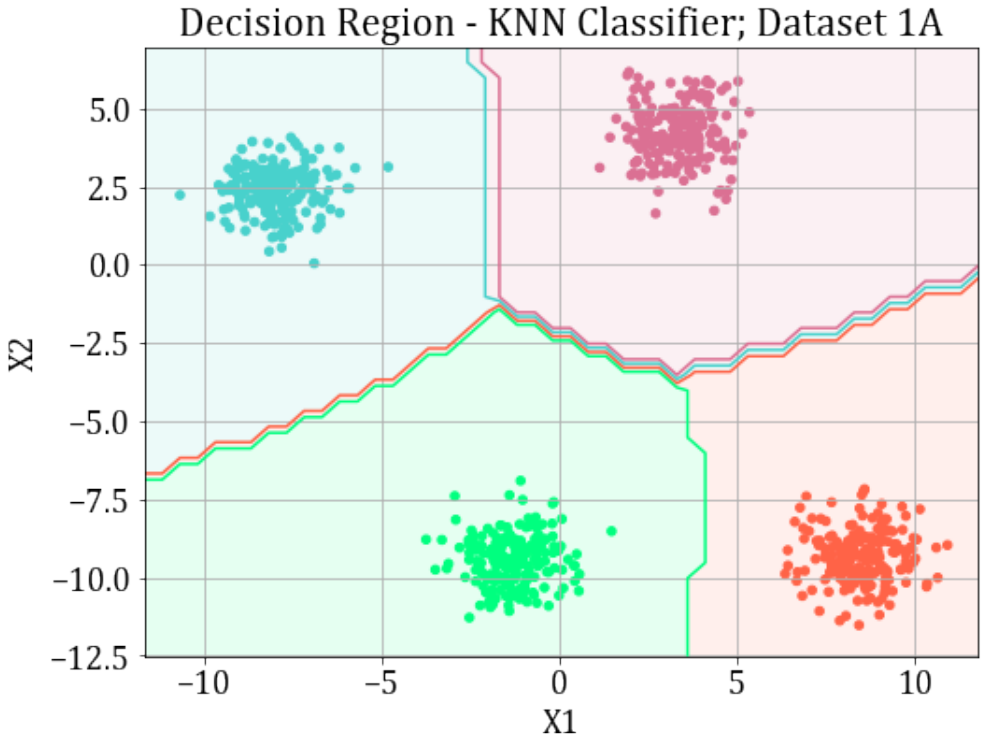


Figure 2: Decision region superimposed with training dataset

The decision boundary obtained (with $k = 1$) is linear in form.

1.2 Naive-Bayes classifier with a Gaussian distribution for each class

1.2.1 Mathematical Formulation

The Bayes Classifiers are probabilistic classifiers based on the Bayes theorem:

$$p(y_i|\vec{x}) = \frac{p(\vec{x}|y_i) * p(y_i)}{p(\vec{x})} \tag{2}$$

Here,

- $p(y_i)$: prior probability for $y = y_i$.
- $p(\vec{x}|y_i)$: Class conditional probability density function or class conditional likelihood function.
- $p(y_i|\vec{x})$: Posterior probability for $y = y_i$ given $\vec{x}$
- $p(\vec{x})$: Evidence or normalization factor

Equation 2 can be re-written as:

$$p(y_i|\vec{x}) = \frac{p(\vec{x}|y_i) \times p(y_i)}{\sum_i p(\vec{x}|y_i) \times p(y_i)} \tag{3}$$

Hence, the probability that $\vec{x}$ belongs to the class y_i is $\propto p(\vec{x}|y_i) \times p(y_i)$.

Step-wise approach:

- $p(y_i)$ is calculated from the train dataset, for dataset 1A, we find that all the classes have equal prior probability.
- The probability $p(\vec{x}|y_i)$ can be calculated by various parametric and non-parametric means. For dataset 1A, we use parametric means as described later.
- $p(y_i|\vec{x})$ is calculated for all the classes using Equation 3.
- The class label with maximum posterior probability is chosen as the class label for $\vec{x}$.

For the discussion that follows for dataset 1A, we assume that $p(\vec{x}|y_i)$ is given by a gaussian distribution:

$$p(\vec{x}|y_i) = \frac{1}{(2\pi)^{d/2} * |C_i|^{1/2}} \exp\left(\frac{-(\vec{x} - \vec{\mu_i})^T * C_i^{-1} * (\vec{x} - \vec{\mu_i})}{2}\right) \tag{4}$$

In the above equation:

- μ_i is the mean corresponding to examples in the class y_i, its dimension is $(d, 1)$, where d is the number of features. Hence, if there are k classes, number of parameters to be estimated for mean: $k * d$.
- C_i is the (d, d) covariance matrix corresponding to the class y_i. Since it is symmetric, number of parameters to be calculated per class: $\frac{d(d+1)}{2}$. Total parameters for covariance: $\frac{k*d(d+1)}{2}$.

The co-variance matrix and eigen-vectors for each class are calculated and are found to be as follows:

- For class y=0.0:

$$C = \begin{bmatrix} 0.676 & -0.005 \\ -0.005 & 0.784 \end{bmatrix}$$

$$eigen - vectors = \begin{bmatrix} -0.999 & 0.049 \end{bmatrix}, \begin{bmatrix} 0.106 & 0.996 \end{bmatrix}$$

- For class y=1.0:

$$C = \begin{bmatrix} 0.731 & 0.022 \\ 0.022 & 0.523 \end{bmatrix}$$

$$eigen - vectors = \begin{bmatrix} 0.994 & -0.106 \end{bmatrix}, \begin{bmatrix} 0.106 & 0.994 \end{bmatrix}$$

- For class y=2.0:

$$C = \begin{bmatrix} 0.767 & 0.009 \\ 0.009 & 0.671 \end{bmatrix}$$

$$eigen - vectors = \begin{bmatrix} 0.996 & -0.090 \end{bmatrix}, \begin{bmatrix} -0.090 & 0.996 \end{bmatrix}$$

- For class y=3.0:

$$C = \begin{bmatrix} 0.705 & 0.028 \\ 0.028 & 0.621 \end{bmatrix}$$

$$eigen - vectors = \begin{bmatrix} 0.957 & -0.289 \end{bmatrix}, \begin{bmatrix} 0.289 & 0.957 \end{bmatrix}$$

1.2.2 Accuracy table and Confusion Matrix

We obtain that irrespective of our assumption of the co-variance matrices, the accuracy over train, validation and test set is 1.

Condition	Train Accuracy	Validation Accuracy	Test Accuracy
$C_i = C_j = \sigma^2 I$	100	100	100
$C_i = C_j = C$	100	100	100
$C_i \neq C_j$	100	100	100

Table 2: Accuracy table for dataset 1A: Bayes Classifier

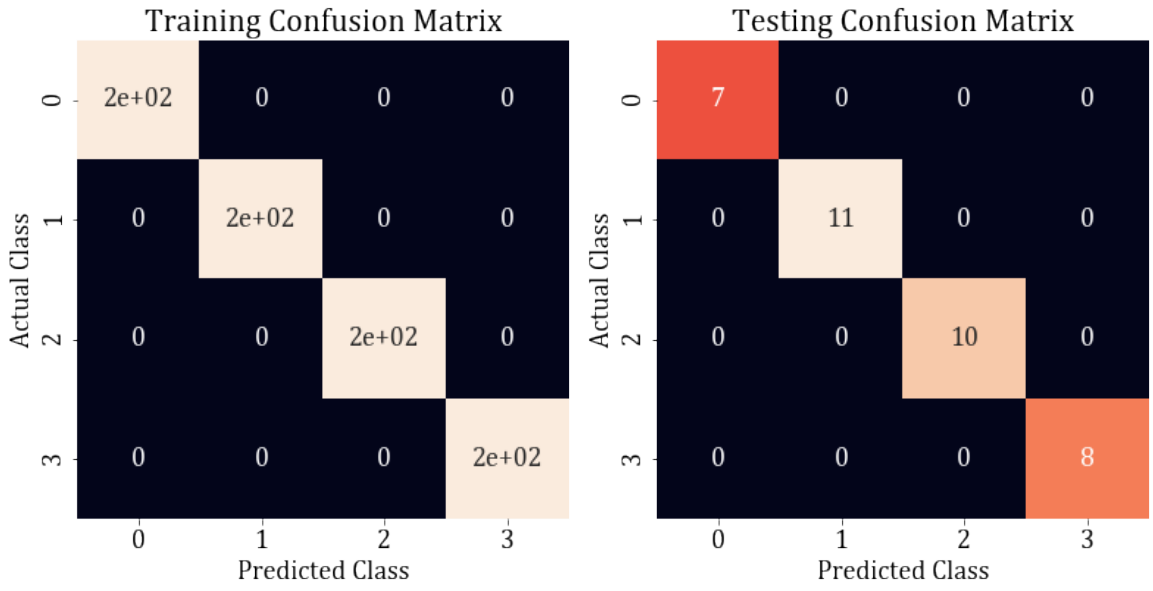


Figure 3: Confusion matrix for train and test data respectively

1.2.3 Level curve plots for different cases

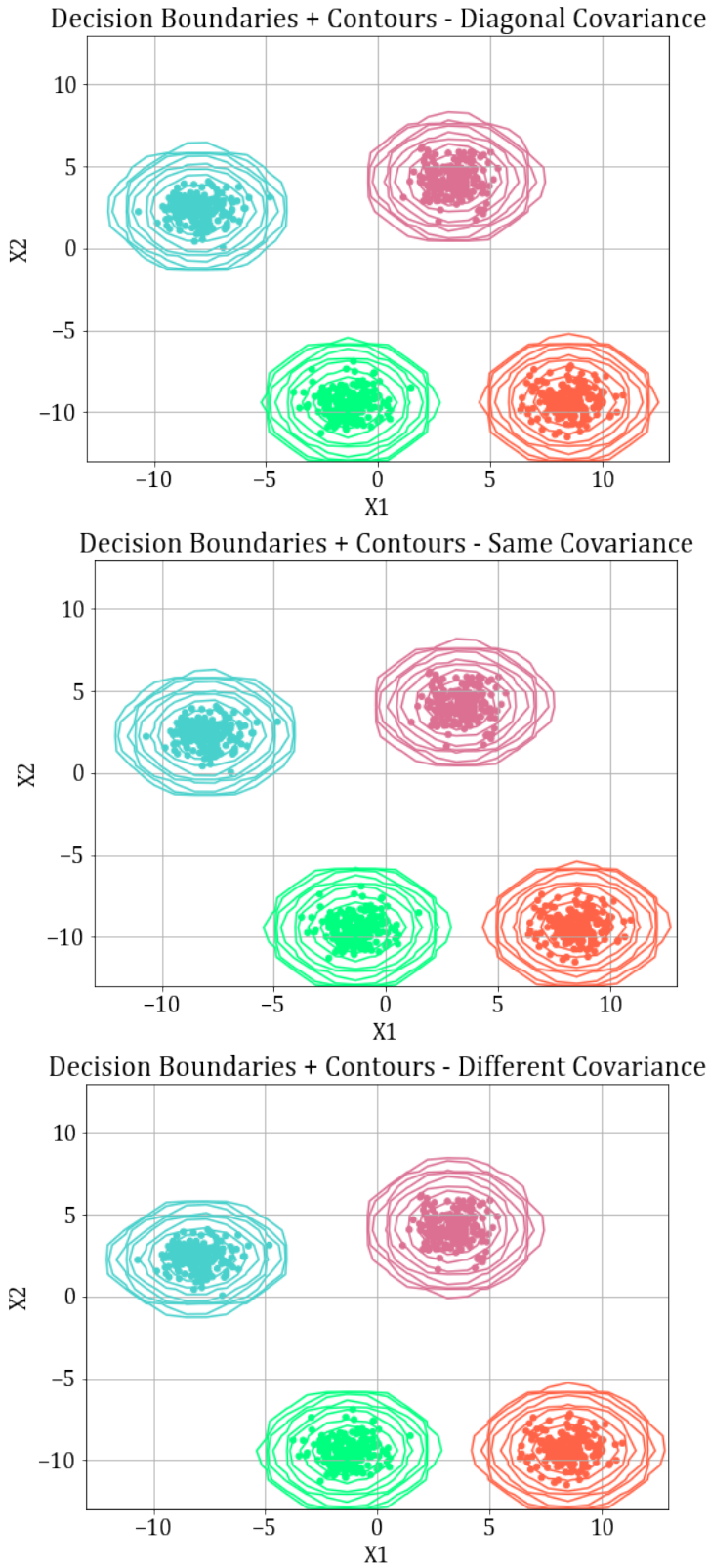


Figure 4: Contour plots

1.2.4 Observations

- For all the three cases, we find that the principal axes of the ellipses are nearly parallel to the coordinate axis. This is because the eigen-vectors are nearly equal to [1,0] and [0,1].
- In case a and case b, the length of principal axes are same across all classes.
- In case c, since we have used different co-variance matrix, the length of principal axes are scaled

accordingly for all classes.

1.2.5 Decision Boundary plot for Naive Bayes Classifier

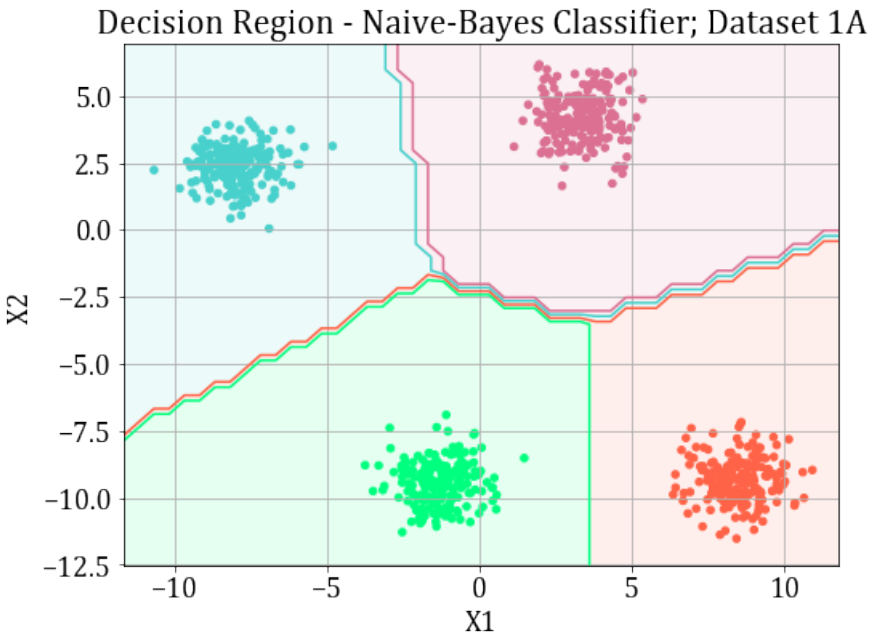


Figure 5: Decision boundary plot

2 Dataset 1B

2.1 K Nearest Neighbor Classifier

Similar to subsubsection 1.1.1, K Nearest Neighbor classifier is used to predict class labels for dataset 2A.

The dataset 1B has 3 unique class labels - $[0.0, 1.0, 2.0]$ and non-linearly separable. The train dataset is of dimension $(800, 3)$ while the dev dataset is of dimension $(90, 3)$. Similar preprocessing steps are performed as in subsubsection 1.1.2.

2.1.1 Model performance across k

Unlike dataset 1A, we observe that the accuracy over validation set decreases on increasing the value of hyper-parameter k. This happens because the dataset 1B is non-linear, a higher k value includes points from other class labels resulting in mis-judgment.''

The accuracy table is as follows:

k-value	Train Accuracy	CV Accuracy	Test Accuracy
1	100.0	100.0	100.0
7	99.5	100.0	97.8
15	99.5	100.0	97.8

Table 3: Accuracy table for data set 2A- knn classifier

Hence, the best configuration of the model exists for $k = 1$.The confusion matrix in this case is:

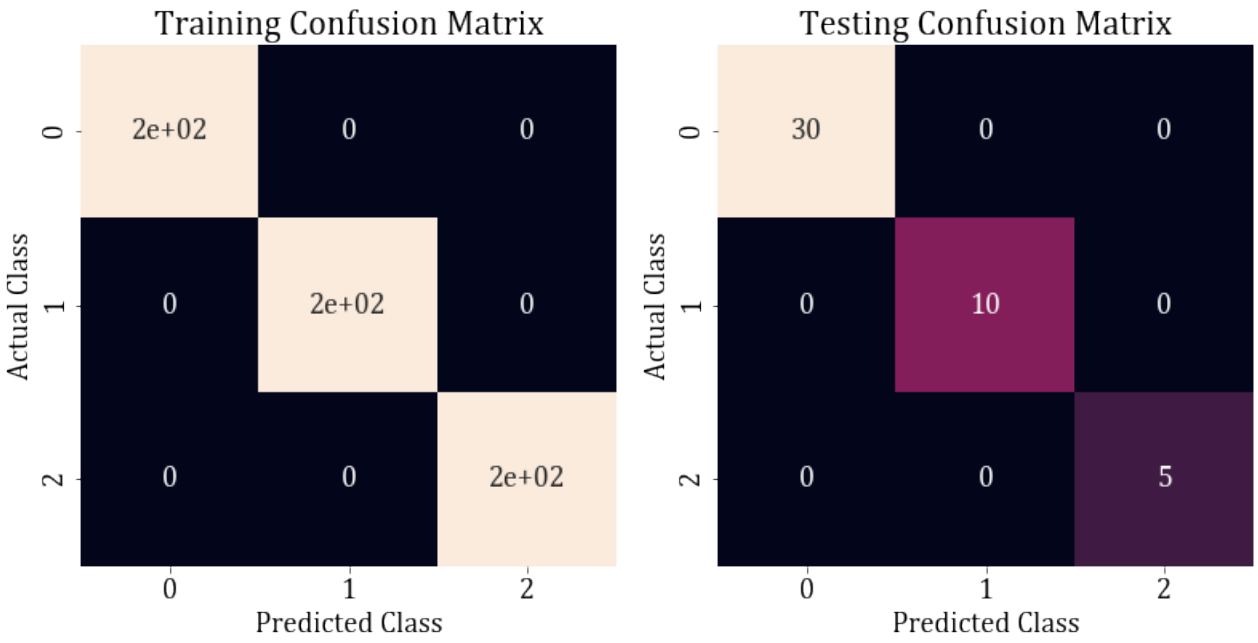


Figure 6: Confusion matrix for $k = 1$, for Train and Test data on left and right respectively

2.1.2 Decision Boundary plot

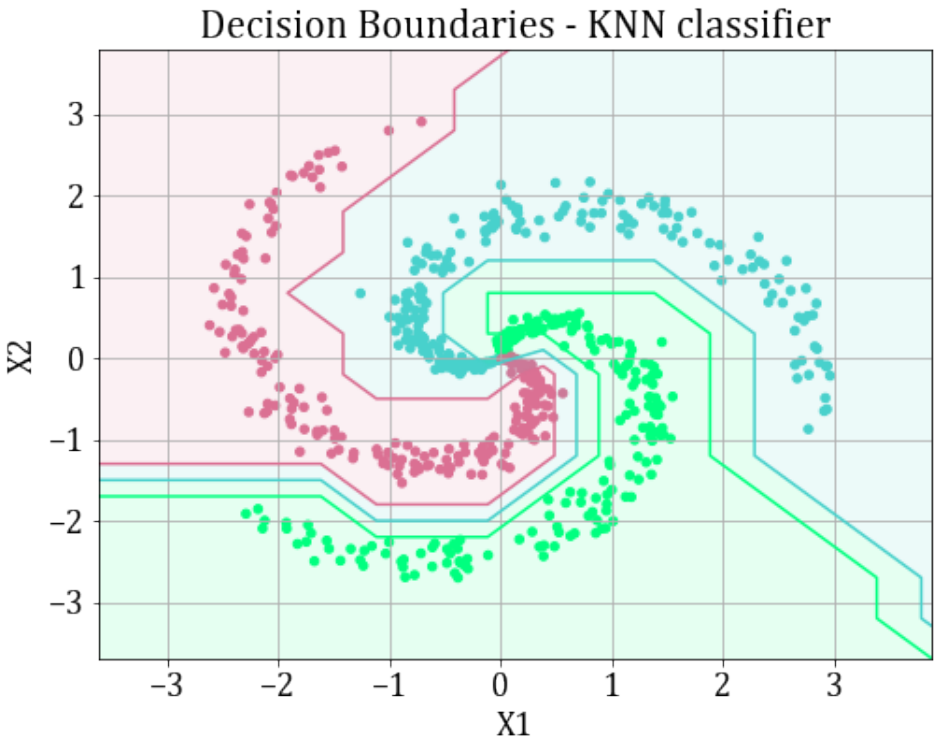


Figure 7: Decision boundary plot for $k = 1$.

The decision boundary obtained is non-linear in form and not smooth.

2.2 Bayes Classifier, GMM, Full Covariance

2.2.1 Equations

Th initialization is done as follows for each class:

- Cluster initialization is done using `kmeans` clustering.
- The relative number of points in each cluster N_q and weightage w_q for each cluster is calculated.
- The responsibility $\gamma_{n,q}$ is then calculated, followed by mean μ_q and covariance matrix C_q.

The parameters are then updated sequentially through the:

- Expectation-step: $\gamma_{n,q}$ is updated.
- Maximization-step: μ_q, C_q, N_q and w_q are updated.

The stopping criterion used is $\Delta(\text{likelihood}) < \texttt{tol}$. The `tol` we considered is 10^{-5}.

2.2.2 Training and Validation Accuracy

The training and validation accuracies obtained for varying q_i for each class is as follows:

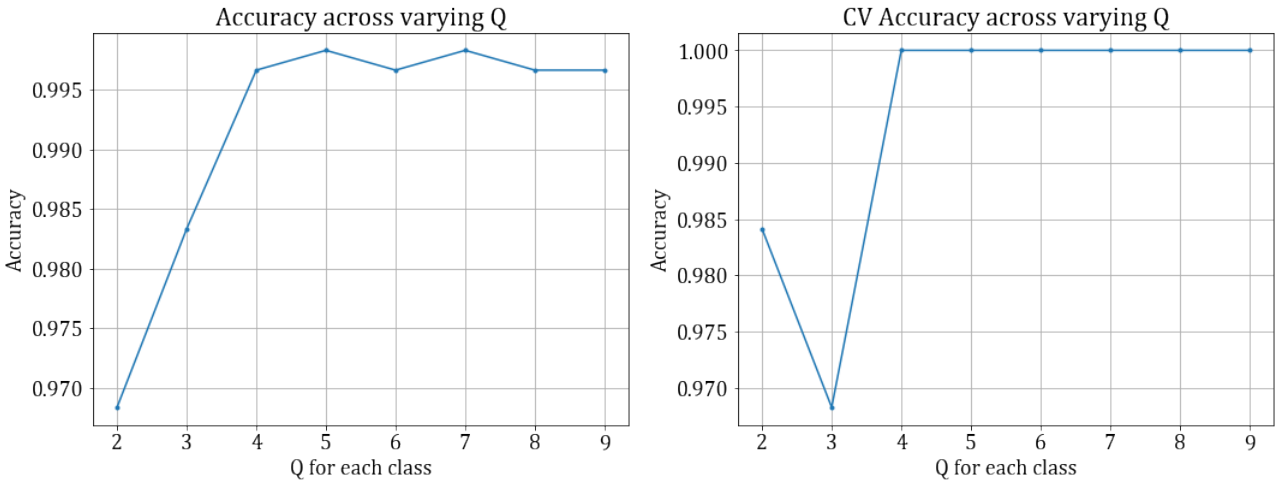


Figure 8: Training and Validation accuracy across q_i, on the left and right respectively, using a GMM model with full covariance matrix.

2.2.3 Testing Accuracy

The testing accuracy obtained for varying q_i for each class is as follows:

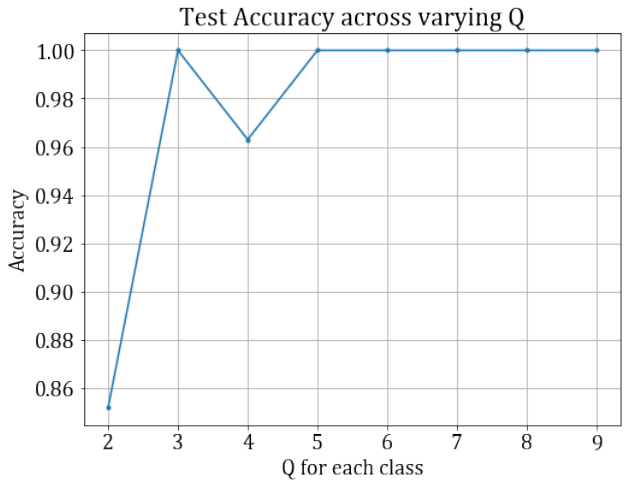


Figure 9: Testing accuracy across q_i, using a GMM model with full covariance matrix.

2.2.4 Best Model

Based on the accuracies obtained on the training, validation and test dataset, the best q_i for the three classes has been chosen as 5. The accuracies obtained in tabular format is as follows:

# Clusters/Class (Q)	Train Accuracy	Validation Accuracy	Test Accuracy
2	0.968333	0.952381	0.925926
3	0.983333	0.968254	1.000000
4	0.996667	0.984127	1.000000
5	0.998333	1.000000	1.000000
6	0.996667	1.000000	1.000000
7	0.998333	1.000000	1.000000
8	0.996667	1.000000	1.000000
9	0.996667	1.000000	1.000000

Table 4: Variation of accuracy across hyperparameter values on the training, validation and test set using full covariance matrix GMM model on Dataset 1B. The row corresponding to the best model has been highlighted.

The confusion matrix obtained for the model with $q_i = 5$ are as follows:

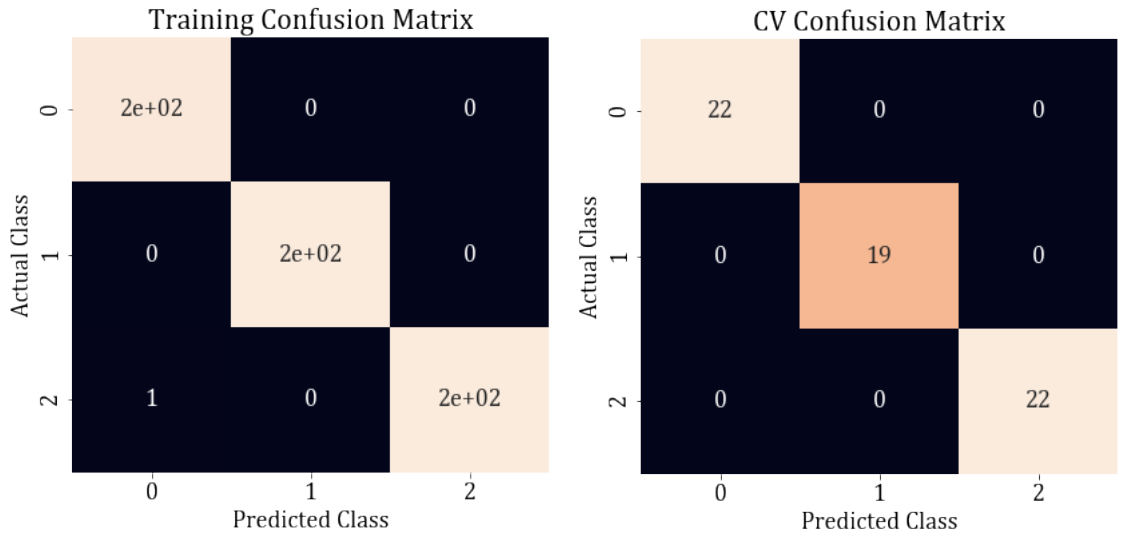


Figure 10: Confusion matrices corresponding to training and validation data, with $q_i = 5$, on the left and right respectively, using GMM model with full covariance.

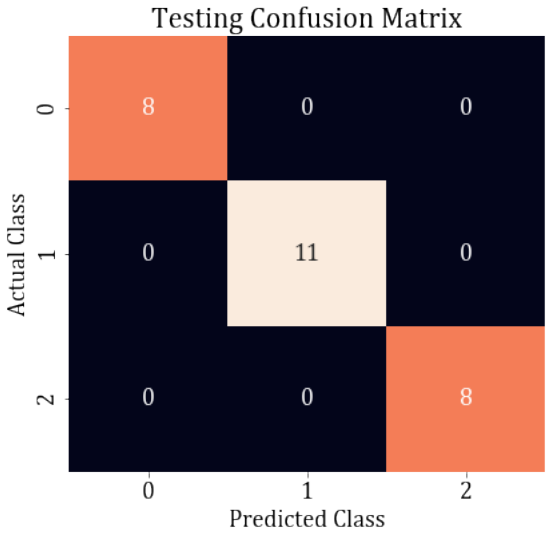


Figure 11: Confusion matrix corresponding to the testing data, with $q_i = 5$, using GMM model with full covariance.

2.2.5 Contour Maps and Decision Surfaces

The contour maps and decision surfaces obtained, with $q_i = 5$ are as follows:

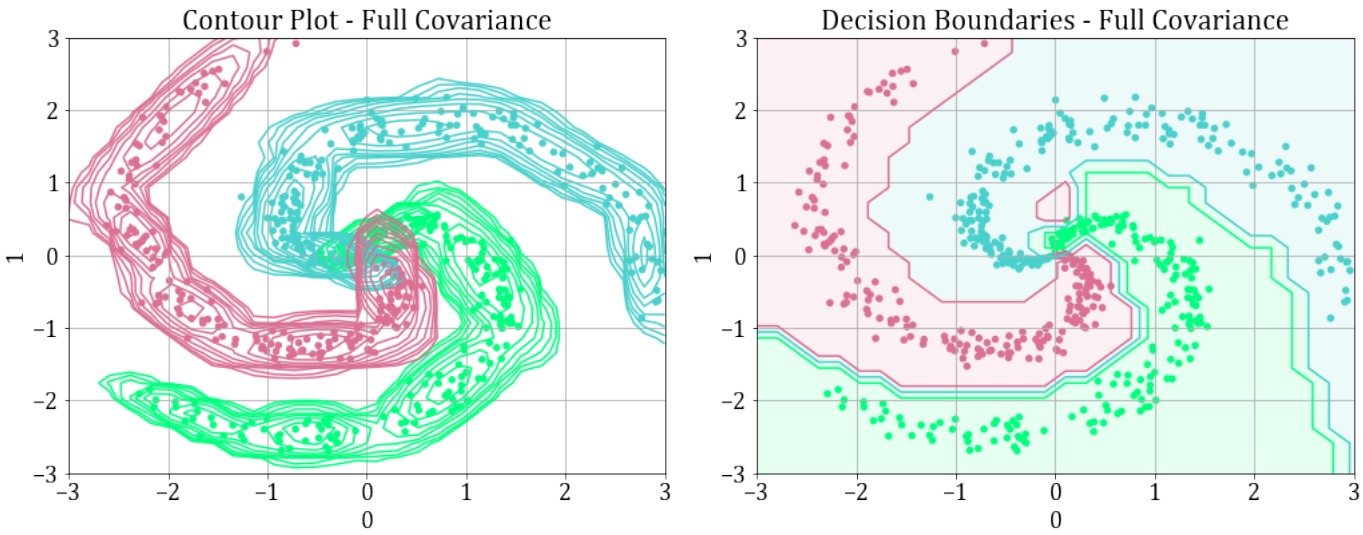


Figure 12: Contour Maps, Decision Surfaces obtained for $q_i = 5$, on the left and right respectively.

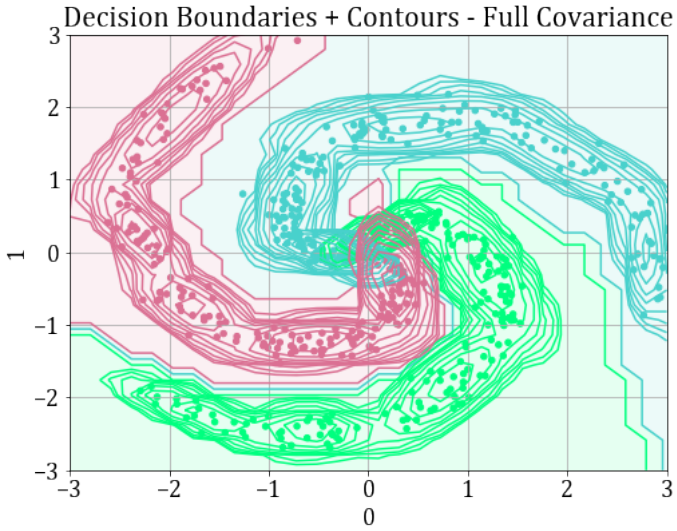


Figure 13: Overlap plot of the decision surface and contours.

2.3 Bayes Classifier, GMM, Diagonal covariance

2.3.1 Training and Validation accuracy

Gaussian multi-modal training (with threshold of the increment in total log-likelihood functions as 0.01) with diagonal covariance matrix was used, for the number of gaussian components Q = 2,3,4,5,6,7,8,9 to estimate the parameters - μ_q, C_q, N_q and w_q for each gaussian component - and predict the classes of the training data (train.csv) and cross-validation (70% of dev.csv). The table 5 shows the results obtained.

# Clusters/Class (Q)	Validation Accuracy	Training Accuracy
2	0.873	0.9166
3	0.920	0.976
4	0.968	0.9966
5	0.984	1.0
6	0.984	0.986
7	0.984	0.991
8	0.984	0.9916
9	0.984	0.9916

Table 5: Variation of accuracy across hyperparameter values on the validation data using the GMM model with diagonal covariance matrix on Dataset 1B. The row corresponding to the best model has been highlighted.

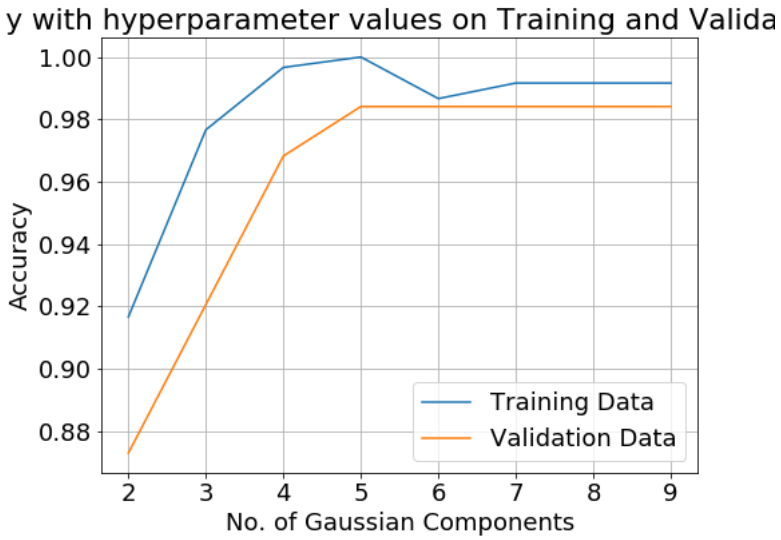


Figure 14: Training and Validation accuracy across q_i, using the GMM model with diagonal covariance matrix on Dataset 1B

2.3.2 Best model output

As we can see in the tables and Figure 14, the best accuracy is when the number of Gaussian components is 5. Using the parameters of the model for 5 gaussian components and predicting for the test dataset (30% of `dev.csv`), the accuracy obtained was **100.0**.
The confusion matrices for the training and test datasets using the best model is Figure 15.

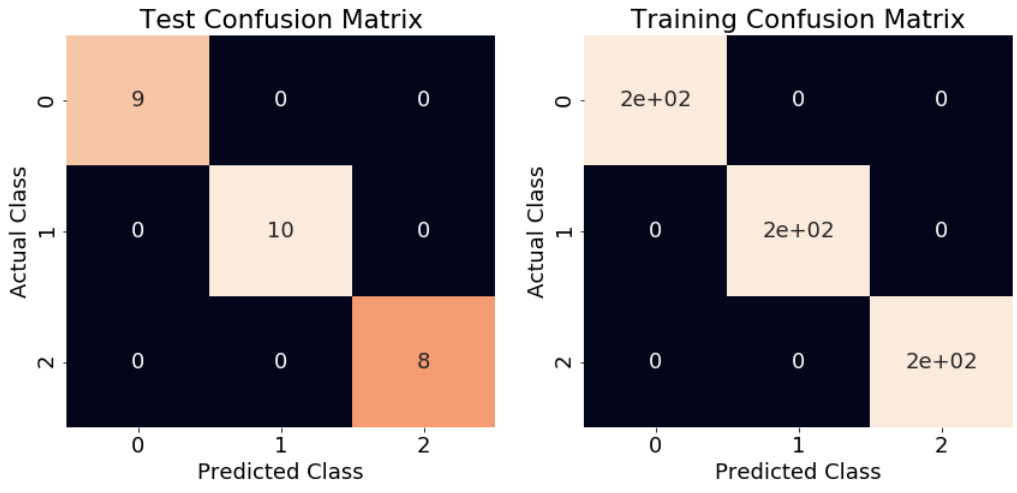


Figure 15: Confusion matrices for data 1B - diagonal covariance matrices

2.3.3 Decision surface

The decision surface plot and the contour plot for the best model is figure 16. As we can see in the decision region plots, the contours follow the non-linear shape of the data. The level curves are ellipses parallel to the coordinate access, since diagonal covariance matrices were used. Figure 16.

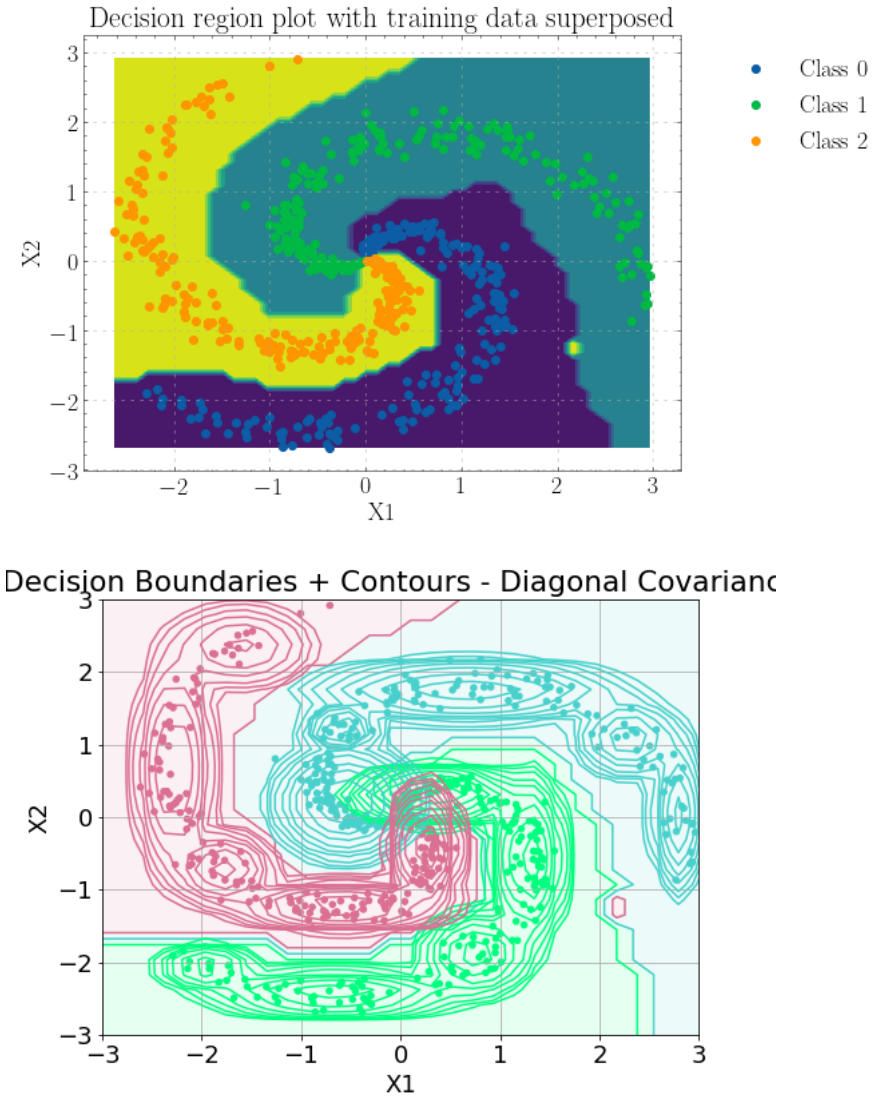


Figure 16: Decision region plot for Bayesian GMM model using diagonal covariance matrix and 5 gaussian components on dataset 1B

2.4 Bayes Classifier with KNN

2.4.1 Mathematical Formulation

While the main principle remains the same as discussed in section (1.2), we now use non-parametric methods to evaluate the class conditional probability $p(\vec{x}|y_i)$.
Suppose the number of data points in the hyper-volume v around $\vec{x}$ be N, the number of data point corresponding to class i be N_i, then:

$$p(\vec{x}|y_i) = \frac{N_i}{N * V} \tag{5}$$

The probability density can be estimated in two ways :

- Specifying the volume V, number of point N_i and N are calculated
- Specifying N, the volume V and N_i are calculated. Here, the radius of hyper sphere is the distance of the point belonging to N farthest from $\vec{x}$, this is called KNN method.

For this case, we use the KNN method to evaluate the class conditional probability densities. The class label i that maximizes Equation 2 is chosen as the label for $\vec{x}$.

2.4.2 Model Performance for varying values of k

The model is tested for $k = 10$ and $k = 20$. The accuracy table and confusion matrix are as follows:

k-value	Train accuracy	validation accuracy	Test Accuracy
10	99.1667	100.0	95.5556
20	98.6667	100.0	93.3333

Table 6: Accuracy table for dataset 2A - Bayes Classifier with knn

Since the accuracy is higher for $k = 10$, it is used to further evaluate the confusion matrix and decision boundary plot.

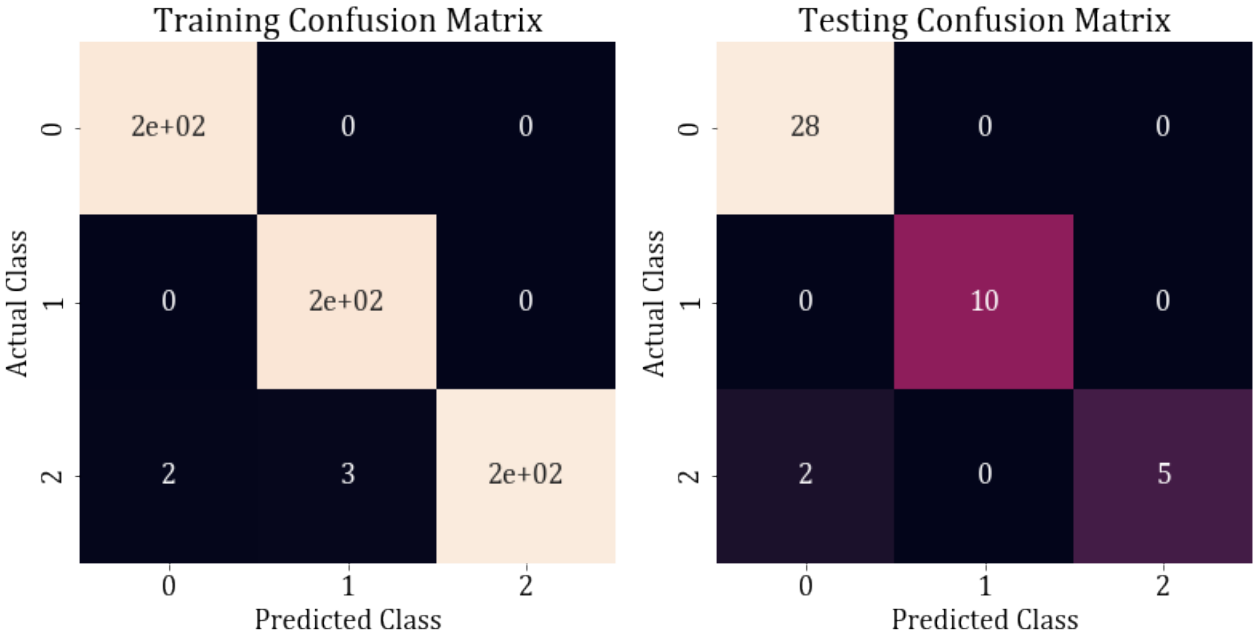


Figure 17: Confusion matrices for $k = 10$, for train data and test data on left and right respectively.

2.4.3 Decision boundary plot

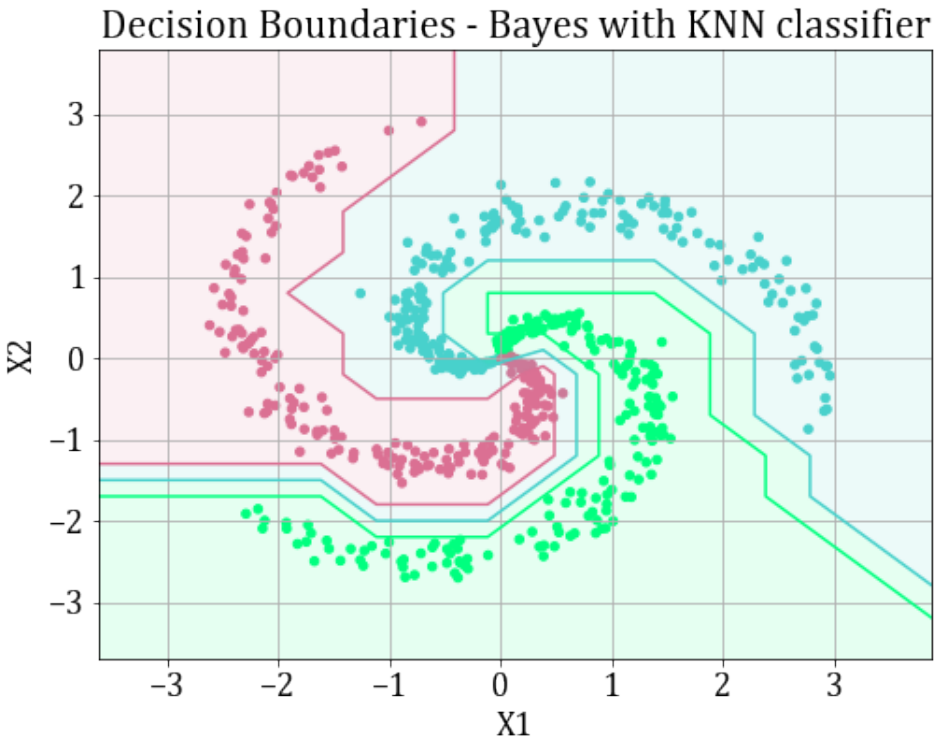


Figure 18: Decision boundary for $k = 10$

While the shape of decision boundary is almost similar as Figure 7, there are still some differences, especially near the edges.

3 Dataset 2A

3.1 Bayes Classifier, GMM, Full Covariance

3.1.1 Training and Validation Accuracy

The training and validation accuracies obtained for varying q_i for each class is as follows:

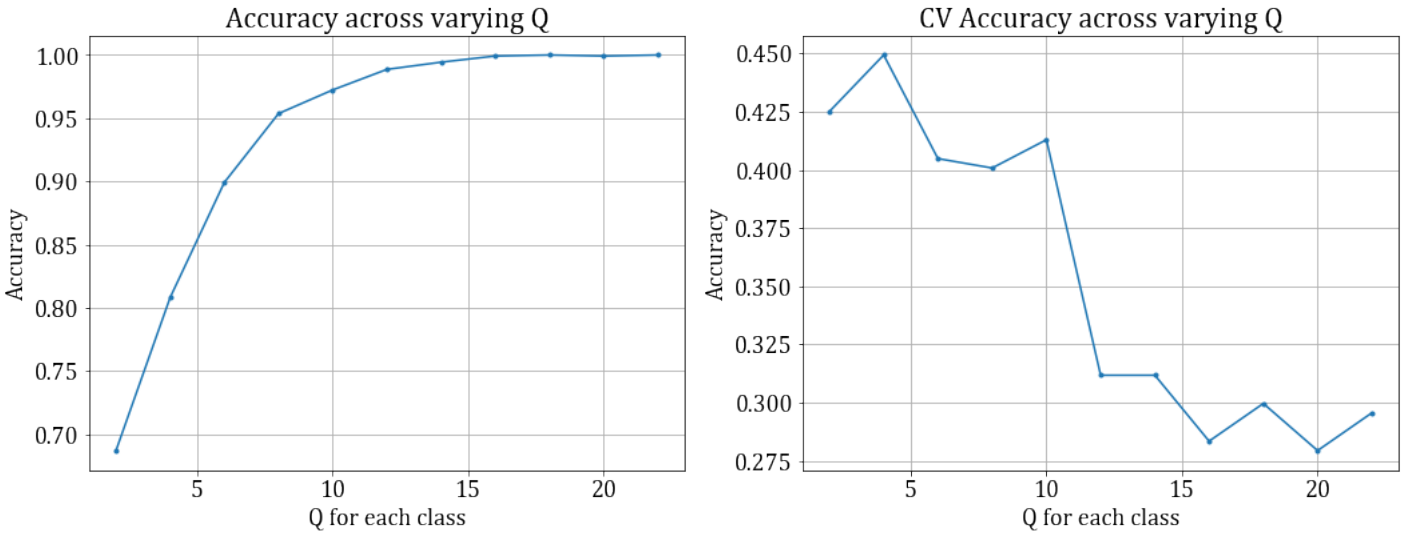


Figure 19: Training and Validation accuracy across q_i, on the left and right respectively

3.1.2 Testing Accuracy

The testing accuracy obtained for varying q_i for each class is as follows:

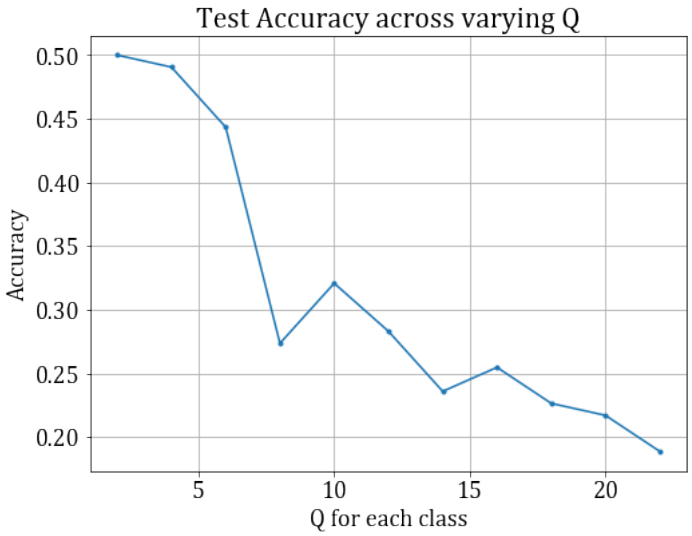


Figure 20: Testing accuracy across q_i

3.1.3 Best Model

Based on the accuracies obtained on the training, validation and test dataset, the best q_i for the three classes has been chosen as 6. The accuracies obtained in tabular format is as follows:

# Clusters/Class (Q)	Train Accuracy	Validation Accuracy	Test Accuracy
4	80.8130	44.9393	49.0566
2	68.6992	42.5101	50.0000
10	97.2358	41.2955	32.0755
6	89.9187	40.4858	44.3396
8	95.3659	40.0810	27.3585
12	98.8618	31.1741	28.3019
14	99.4309	31.1741	23.5849
18	00.0000	29.9595	22.6415
22	00.0000	29.5547	18.8679
16	99.9187	28.3401	25.4717
20	99.9187	27.9352	21.6981

Table 7: Accuracy across hyperparameter values on the training, validation and test dataset using the GMM model with full covariance matrix on Dataset 2A.

The confusion matrix obtained for the model with $q_i = 6$ are as follows:

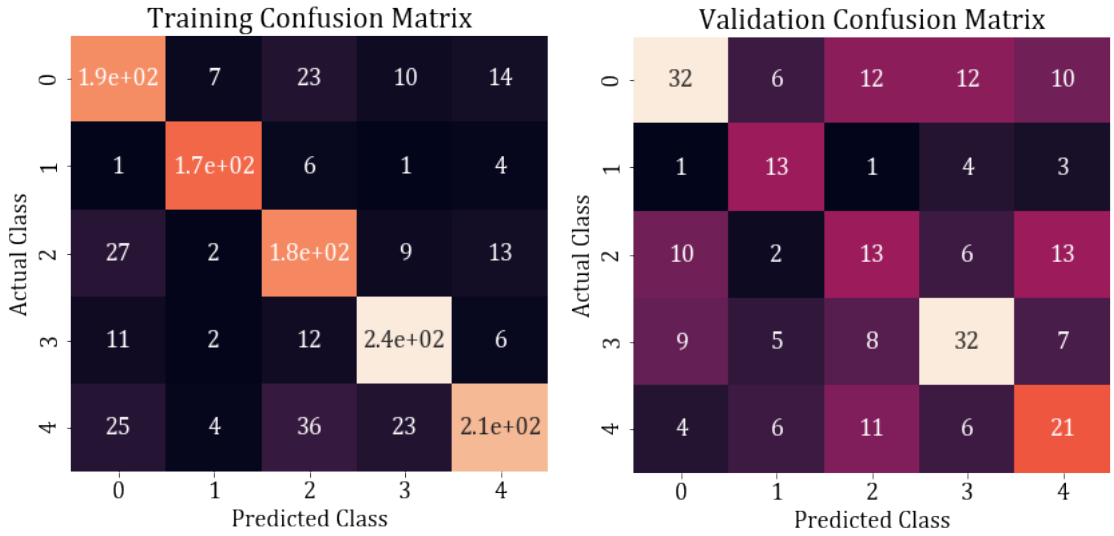


Figure 21: Training and Validation confusion matrices for the best model with $q_i = 6$, on the left and right respectively

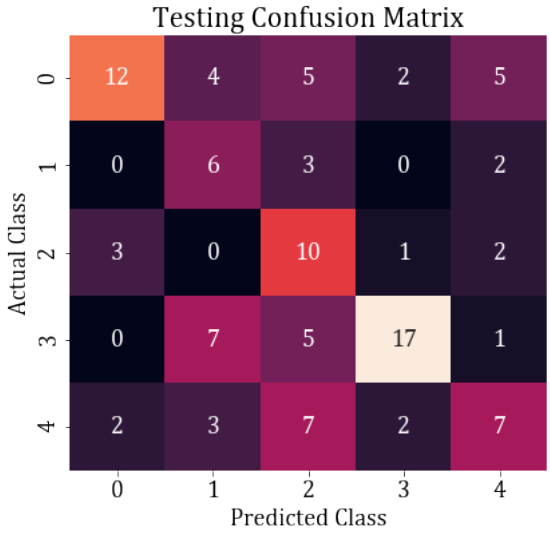


Figure 22: Testing confusion matrix for the best model with $q_i = 6$.

In addition to just taking the same number of clusters for all classes, a parameter sweep was done to identify the best combination of cluster numbers for the dataset. The accuracies obtained from the parameter sweeps are as follows:

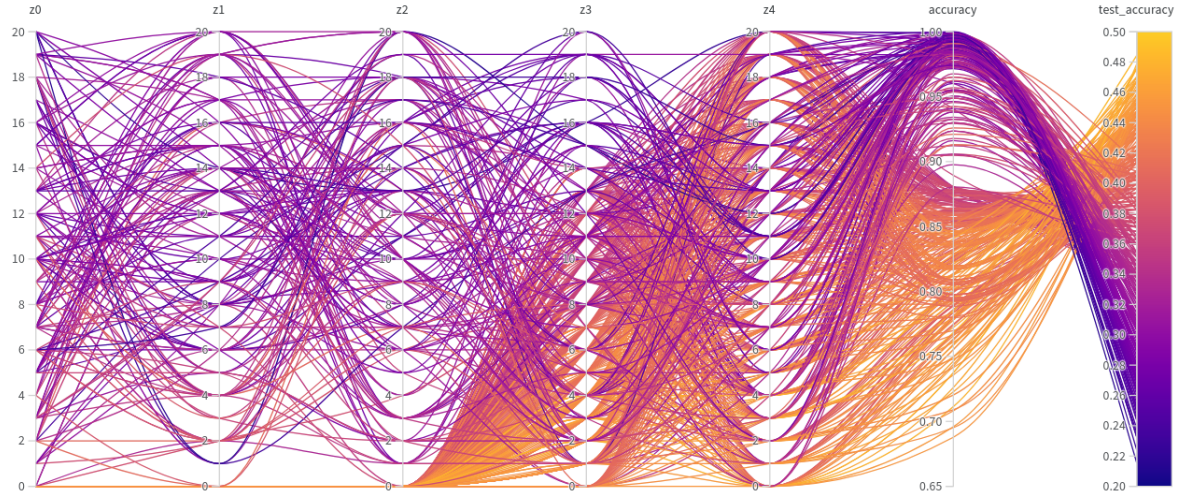


Figure 23: Parameter Sweep Results for the dataset 2A.

From the graph above, the parameter combination that resulted in the best validation accuracy is:

- $q_1 : 2$
- $q_2 : 2$
- $q_3 : 2$
- $q_4 : 6$
- $q_5 : 3$

The accuracies obtained are as follows:

- Training accuracy: 0.7853658536585366
- Validation accuracy: 0.4939271255060729
- Testing accuracy: 0.46226415094339623

From the results above, we can see that the validation accuracy is higher in this case, however, the train and test accuracies are low.

The confusion matrices obtained are as follows:

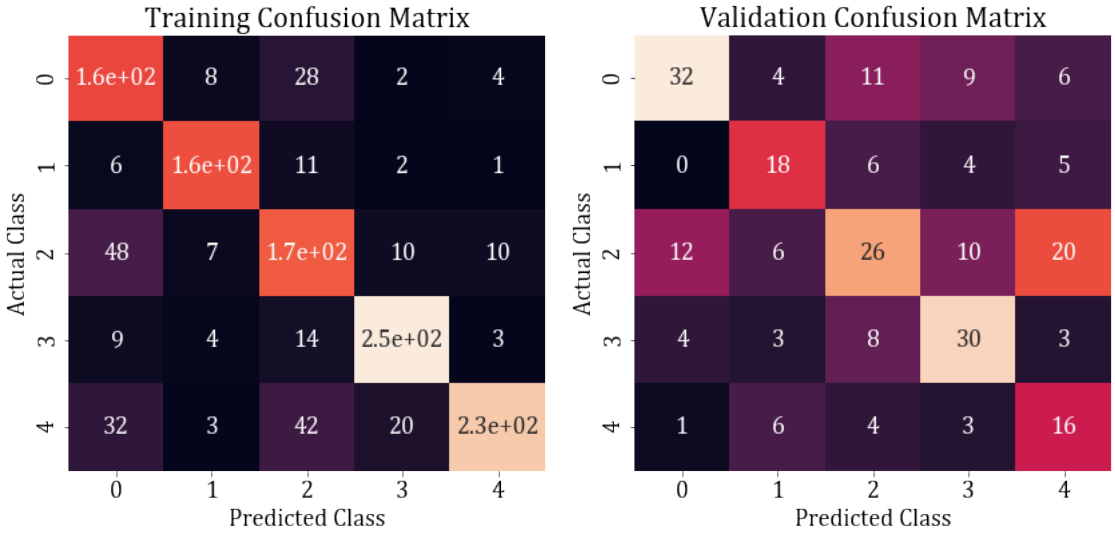


Figure 24: Training and Validation confusion matrices for the model with varying q_i, on the left and right respectively

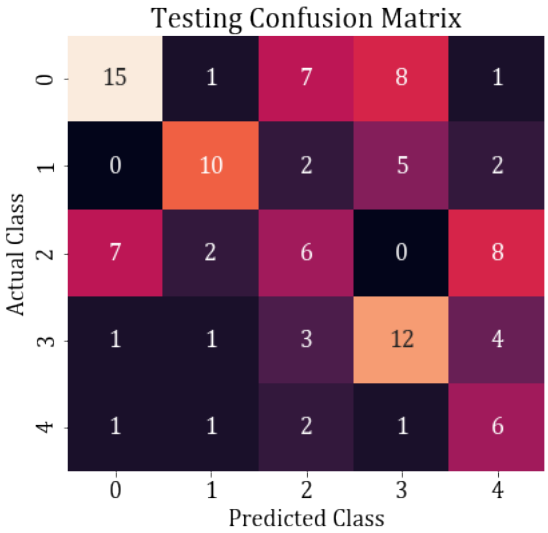


Figure 25: Testing confusion matrix for the model with varying q_i.

3.2 Bayes Classifier, GMM, Diagonal covariance

3.2.1 Training and Validation Accuracy

The accuracy obtained on training the data 2A on GMM model with diagonal covariance matrix is as in Table 8.

# Clusters/Class (Q)	Train Accuracy	Validation Accuracy	Test Accuracy
16	76.9106	46.9636	42.4528
4	57.1545	46.9636	36.7925
20	82.9268	45.7490	41.5094
14	75.5285	44.9393	35.8491
10	68.6179	44.5344	41.5094
8	67.9675	42.5101	33.0189
6	62.8455	41.2955	41.5094
22	83.8211	39.6761	40.5660
18	78.9431	39.6761	41.5094
12	73.9024	39.2713	40.5660
2	49.5122	38.0567	33.9623

Table 8: Accuracy across hyperparameter values on the training, validation and test dataset using the GMM model with diagonal covariance matrix on Dataset 2A.

The training and validation accuracies obtained for varying q_i for each class is as follows:

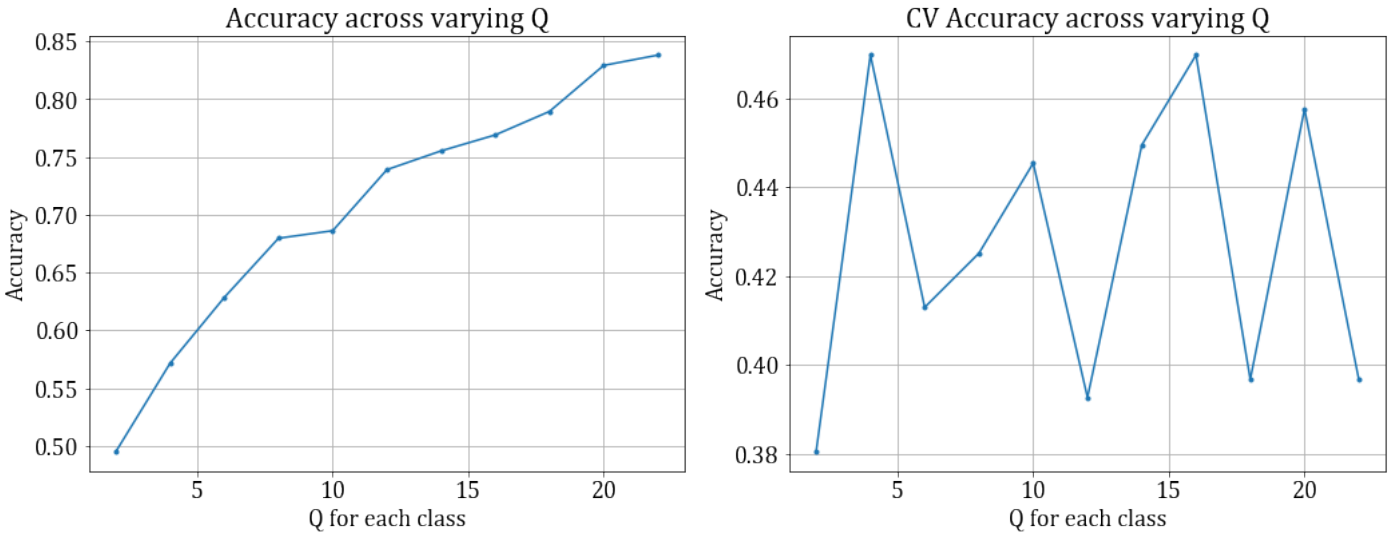


Figure 26: Training and Validation accuracy across q_i, on the left and right respectively

3.2.2 Testing Accuracy

The testing accuracy obtained for varying q_i for each class is as follows:

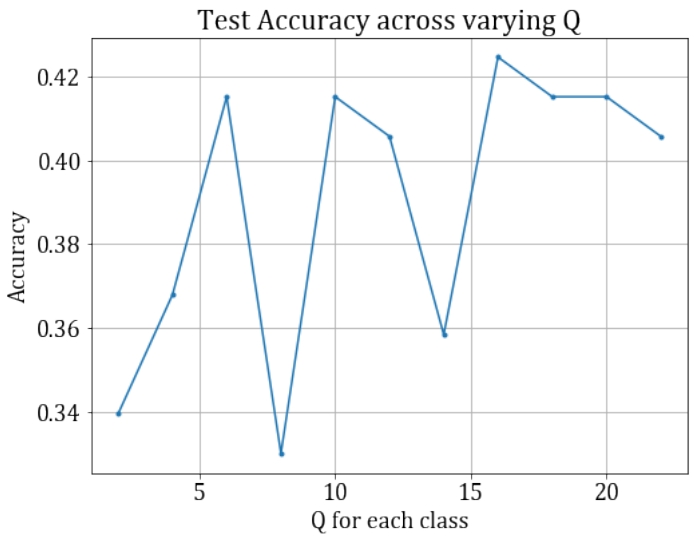


Figure 27: Testing accuracy across q_i

3.2.3 Best model on test data

The highest accuracy on validation dataset is for 16 gaussian components. Applying this model to predict the test data, we get an accuracy of **42.45%**. The confusion matrices for this model on training, validation and test data is as follows:

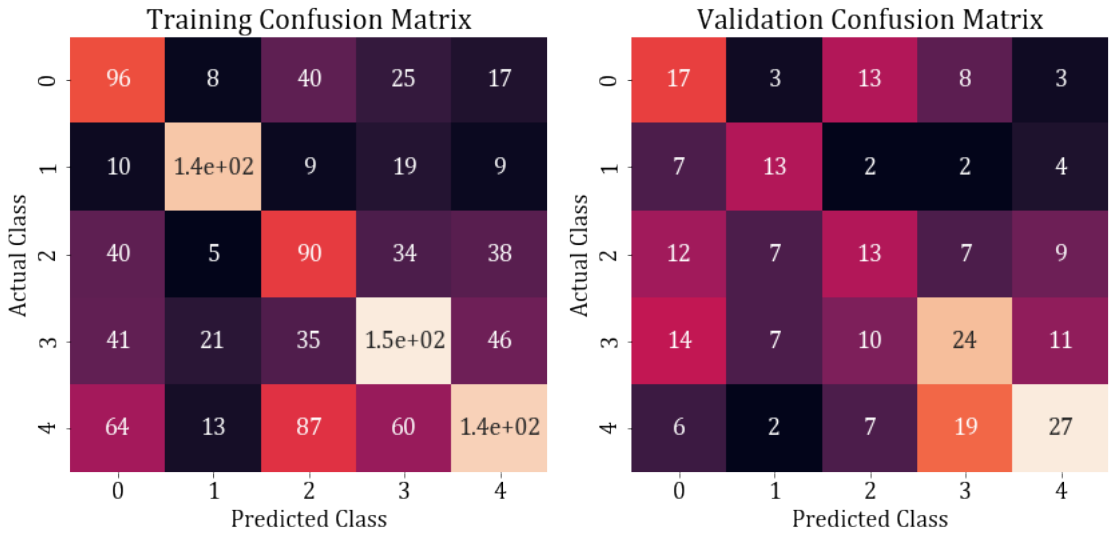


Figure 28: Training and Validation confusion matrices for the best model with $q_i = 16$, on the left and right respectively

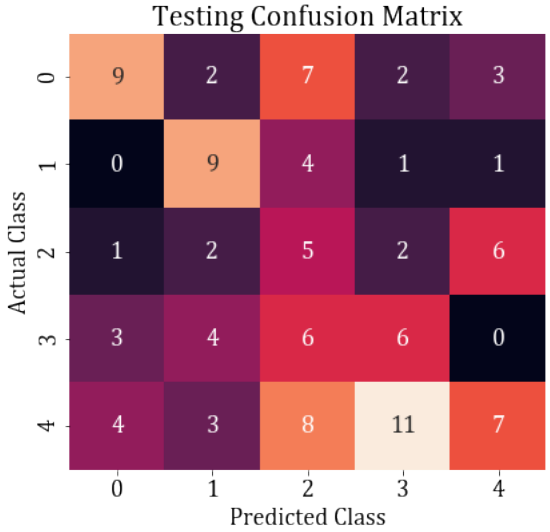


Figure 29: Testing confusion matrix for the best model with $q_i = 16$.

4 Dataset 2B

The varying length classification is done as follows:

- Each image has a $36 * 23$ feature parameters.
- Each of these 36 rows are considered as features and a GMM (full covariance/diagonal covariance) is fit on each of the features.
- The Expectation-step and Maximization-step are performed on the data, as is in case of static length pattern.
- For each of the data points,

$$p(X|\lambda_i) = \prod_{t=1}^{T} p(x_t|\lambda_i) = \prod_{t=1}^{T} \sum_{q=1}^{Q} w_{iq}\mathcal{N}(x_t|\mu_{iq}, \Sigma_{iq})f \tag{6}$$

  is calculated to perform classification.
- In order to circumvent the problem associated with floating point precession, the gaussian probabilities for each of the 36 features were normalized across classes and the product was then performed.
- The data point is then allotted to the class that has the maximum probability.

4.1 Bayes Classifier, GMM, Full Covariance

4.1.1 Training Accuracies

The class-wise training accuracy obtained for varying q's is as follows:

Q	Coast	Highway	Mountain	Open Country	Tall Building
3	99.601594	100	99.616858	100	100
5	100	100	100	100	100
14	100	100	100	100	100
20	100	100	100	100	100

Table 9: Accuracy across hyperparameter values on the training dataset using the GMM model with full covariance matrix on Dataset 2B.

4.1.2 Dev Accuracies

The class-wise accuracy obtained on the dev dataset, for varying q's is as follows:

Q	Coast	Highway	Mountain	Open Country	Tall Building
3	86.301370	46.153846	73.333333	69.512195	90.140845
5	78.082192	32.692308	54.666667	63.414634	91.549296
14	80.821918	0.000000	45.333333	40.243902	92.957746
20	80.821918	0.000000	45.333333	40.243902	92.957746

Table 10: Accuracy across hyperparameter values on the dev dataset using the GMM model with full covariance matrix on Dataset 2B.

From the tables above, we see that the maximum accuracy for the classes: coast, highway, mountain and open country are for 3 GMMs per feature. However, the accuracy for the class tall buildings increases with increasing number of GMM components.

4.1.3 Best Model

The highest accuracy on validation dataset is for 3 gaussian components for each feature. The overall training accuracy for the best model chosen is: 99.83739837398375

Applying this model to predict the test data, we get an accuracy of **74.50%**. The confusion matrices for this model on training, validation and test data is as follows:

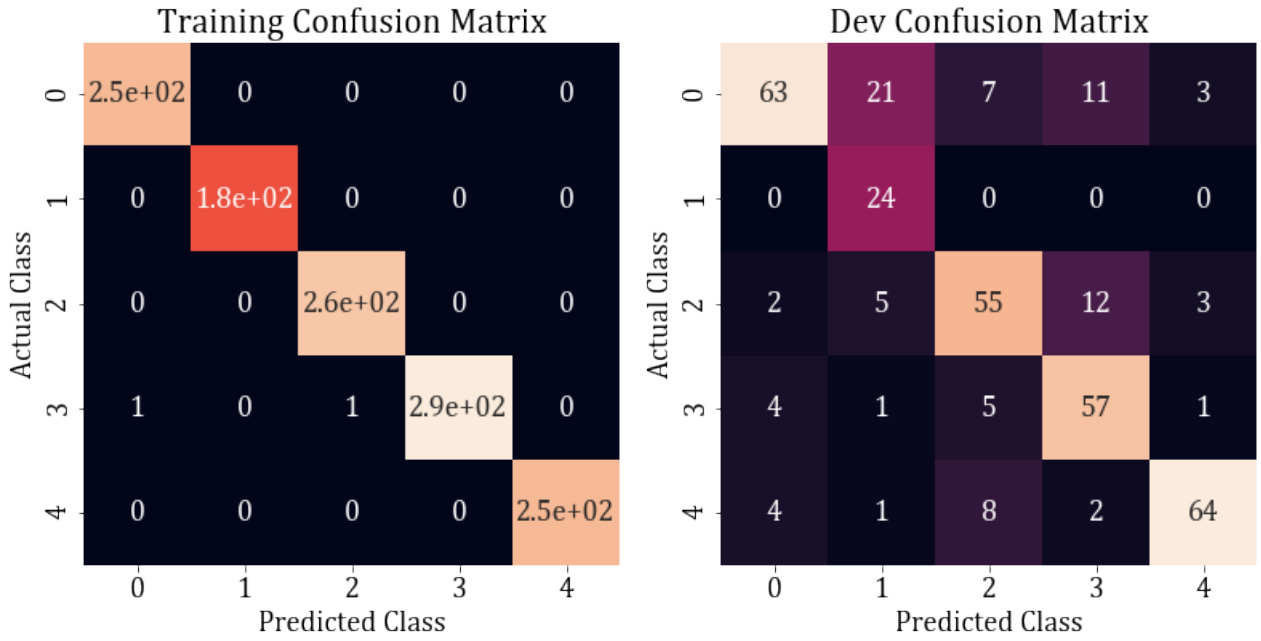


Figure 30: Training and Dev confusion matrices for the best model with $q_i = 3$, on the left and right respectively

4.2 Bayes Classifier, GMM, Diagonal Covariance

4.2.1 Training Accuracies

The class-wise training accuracy obtained for varying q's is as follows:

Q	Coast	Highway	Mountain	Open Country	Tall Building
3	85.657371	87.912088	88.505747	90.592334	88.755020
5	92.828685	95.604396	94.636015	95.121951	95.582329
14	99.203187	100	100	100	99.196787
20	100	100	100	100	99.598394

Table 11: Accuracy across hyperparameter values on the training dataset using the GMM model with diagonal covariance matrix on Dataset 2B.

4.2.2 Dev Accuracies

The class-wise accuracy obtained on the dev dataset, for varying q's is as follows:

Q	Coast	Highway	Mountain	Open Country	Tall Building
3	61.643836	65.384615	70.666667	78.048780	76.056338
5	68.493151	69.230769	77.333333	74.390244	77.464789
14	78.082192	48.076923	66.666667	74.390244	87.323944
20	75.342466	46.153846	73.333333	71.951220	90.140845

Table 12: Accuracy across hyperparameter values on the training dataset using the GMM model with diagonal covariance matrix on Dataset 2B.

From the tables above, we see that the maximum accuracy for the classes: coast, highway, mountain and open country are for 5 GMMs per feature. However, the accuracy for the class tall buildings increases with increasing number of GMM components.

4.2.3 Best Model

The highest accuracy on validation dataset is for 5 gaussian components for each feature. The overall training accuracy for the best model chosen is: 94.71544715447155

Applying this model to predict the test data, we get an accuracy of **73.65%**. The confusion matrices for this model on training, validation and test data is as follows:

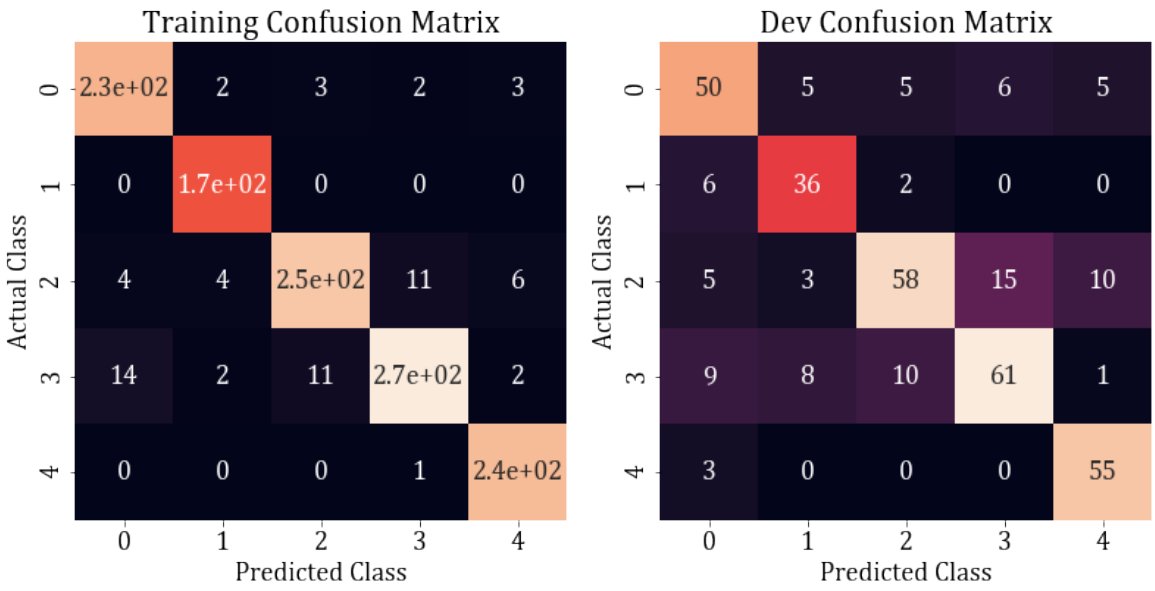


Figure 31: Training and Dev confusion matrices for the best model with $q_i = 5$, on the left and right respectively

4.3 Inference

From Table 9, Table 10, Table 11 and Table 12, we can observe the follows:

- The training accuracy increases as the number of components per feature increases irrespective of the type of covariance matrix.
- For the classes: Coast and Tall Building, full covariance models better fit the data than the diagonal covariance models. This indicates that the features have variations in directions that aren't parallel to the coordinate axis.
- For the classes: highway, mountain and open country, diagonal covariance models better fit the data. Additionally we also note that that accuracy doesn't increase indiscriminately as a function of q_i, but that it peaks at an intermediate value. This could mean that the features have variations in the direction parallel to the coordinate axis and that the complexity of the data is best explained by an intermediate number of components per feature.