CS5691: Programming Assignment 2

Team 20

Niveath A CS21B060

Shaun Mathew CS21B076

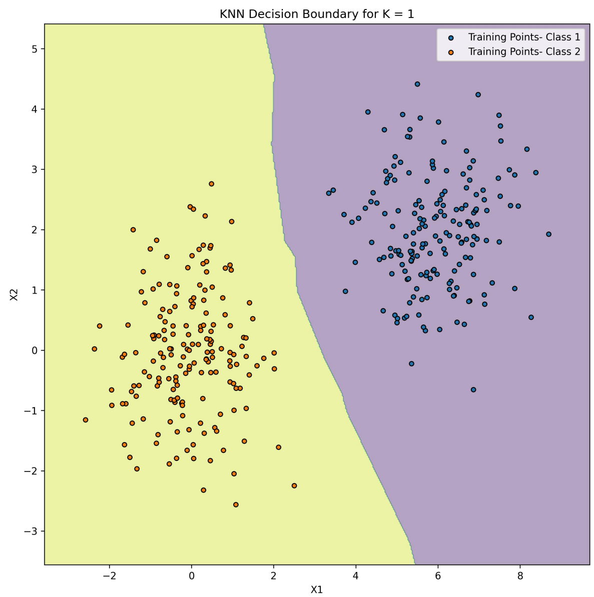
October 29, 2023

Contents

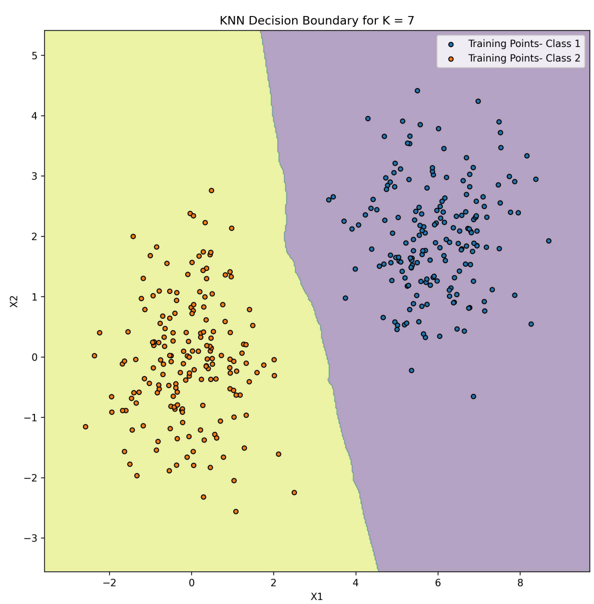
1 Dataset 1(a)

1.1 K-Nearest Neighbours Classifier

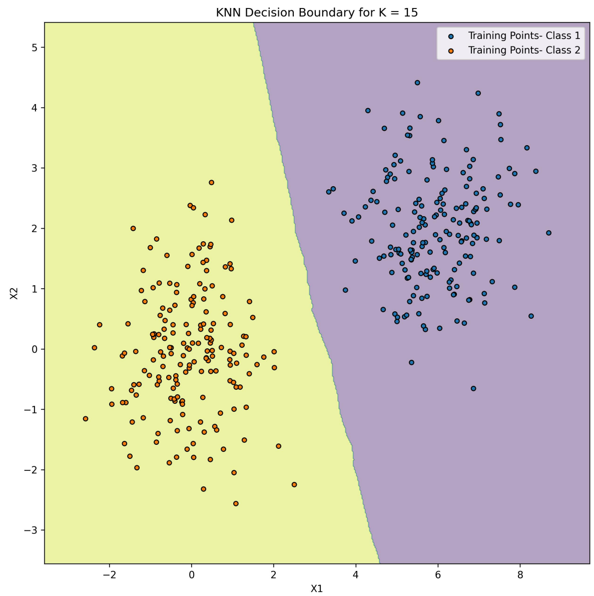
1.1.1 Plots of the Decision Surface



(1) $K = 1$



(2) $K = 7$



(3) $K = 15$

Figure 1: Decision Surface of the Classifier for different K

1.1.2 Hyperparameters v/s Training, Validation and Testing Accuracy

K	Training Accuracy	Validation Accuracy	Testing Accuracy
1	1.0	1.0	1.0
7	1.0	1.0	1.0
15	**1.0**	**1.0**	**1.0**

Table 1: Training, Validation and Testing Accuracy for different K

1.1.3 Inferences

1. Upon scrutinizing the distribution of training data, it becomes evident that distinct patterns emerge, suggesting clear linear separability. Consequently, the aptness of the K-Nearest Neighbours algorithm for computing class-conditional probabilities in this dataset is apparent.

2. The model's robust performance is underscored by its impressive classification accuracy, even when employing a relatively large K value of 15. This accuracy is consistently observed across training, validation, and testing datasets.

3. Opting for $K = 15$ is a strategic decision aimed at averting overfitting, a common challenge with smaller K values. By considering a broader set of neighboring data points, our model strikes a balance, demonstrating a nuanced approach to class-conditional probability computation.

4. Conversely, larger K values introduce a smoothing effect, potentially leading to an underfit model with a more generalized decision boundary. Striking the right balance is essential.

5. It is imperative to acknowledge the intricacies involved in determining the optimal K value. The absence of an expansive dataset for comprehensive model testing presents a notable challenge in achieving this precision.

1.1.4 Confusion Matrix for the best classifier

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	182	0
Class 1	0	178

Table 2: Confusion Matrix for training dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	47	0
Class 1	0	43

Table 3: Confusion Matrix for validation dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	21	0
Class 1	0	29

Table 4: Confusion Matrix for testing dataset

1.2 Naive Bayes Classifier using a Gaussian Distribution with equal covariance matrices

1.2.1 Plot of the Decision Surface and Level Curves

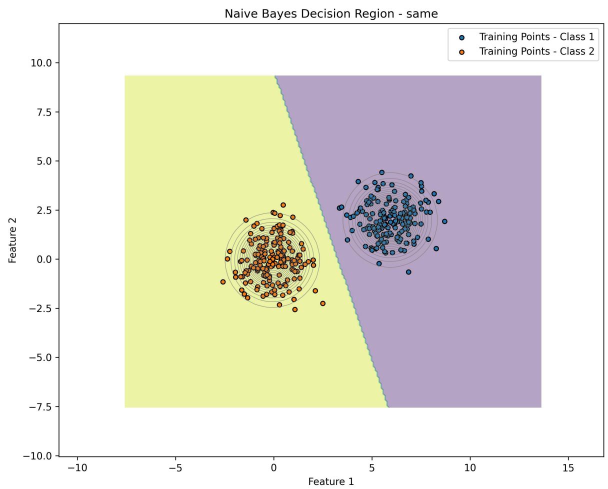


Figure 2: Decision Surface of the Classifier and Level Curves of the Gaussian Distribution

1.2.2 Training, Validation and Testing Accuracy

Training Accuracy	Validation Accuracy	Testing Accuracy
1.0	1.0	1.0

Table 5: Training, Validation and Testing Accuracy of the classifier.

1.2.3 Inferences

1. Applying a Naive Bayes Gaussian estimator with a shared covariance matrix to our linearly separable data resulted in a highly accurate model across all datasets.
2. The straightforward nature of this approach, coupled with its effective classification, aligns well with the inherent linear separability of the data.
3. Despite assuming a common covariance matrix, the model maintains proficiency in discerning class-conditional probabilities.
4. Our parameter choice balances model simplicity with capturing essential characteristics of the linearly separable classes.
5. Optimal parameter determination, particularly in the absence of an extensive dataset for comprehensive testing, underscores the need for a judicious configuration approach.

1.2.4 Confusion Matrix for the classifier

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	182	0
Class 1	0	178

Table 6: Confusion Matrix for training dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	47	0
Class 1	0	43

Table 7: Confusion Matrix for validation dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	21	0
Class 1	0	29

Table 8: Confusion Matrix for testing dataset

1.3 Naive Bayes Classifier using a Gaussian Distribution with different covariance matrices

1.3.1 Plot of the Decision Surface and Level Curves

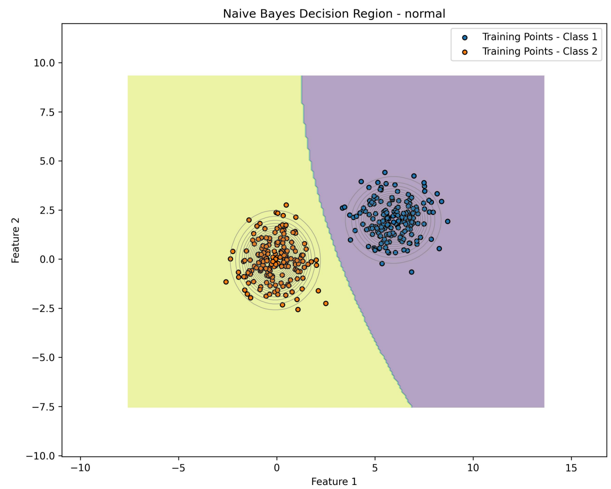


Figure 3: Decision Surface of the Classifier and Level Curves of the Gaussian Distribution

1.3.2 Training, Validation and Testing Accuracy

Training Accuracy	Validation Accuracy	Testing Accuracy
1.0	1.0	1.0

Table 9: Training, Validation and Testing Accuracy of the classifier.

1.3.3 Inferences

1. Applying a Naive Bayes Gaussian estimator to our linearly separable data without assuming a shared covariance matrix maintains high accuracy across training, validation, and testing datasets.
2. Allowing distinct covariance matrices for each class enhances the model's adaptability to the data's intricacies, reflecting in sustained effective classification.
3. The model, now unconstrained by a common covariance matrix, adeptly captures nuanced variations within each class, contributing to its robust performance.
4. Our parameter choice, accounting for distinct covariance matrices, aligns the model with the inherent complexity of the linearly separable classes.
5. The pursuit of optimal parameters remains crucial, emphasizing the need for a thoughtful configuration approach, especially in the absence of an extensive dataset for comprehensive testing.

1.3.4 Confusion Matrix for the classifier

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	182	0
Class 1	0	178

Table 10: Confusion Matrix for training dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	47	0
Class 1	0	43

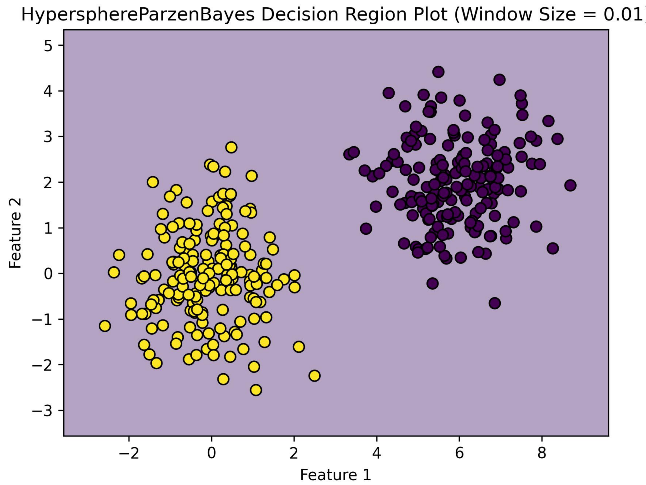
Table 11: Confusion Matrix for validation dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	21	0
Class 1	0	29

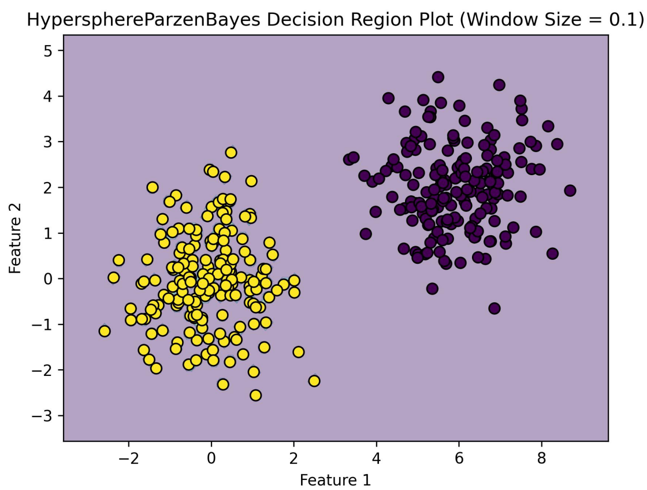
Table 12: Confusion Matrix for testing dataset

1.4 Bayes Classifier with Hypersphere-based Parzen-window density estimation

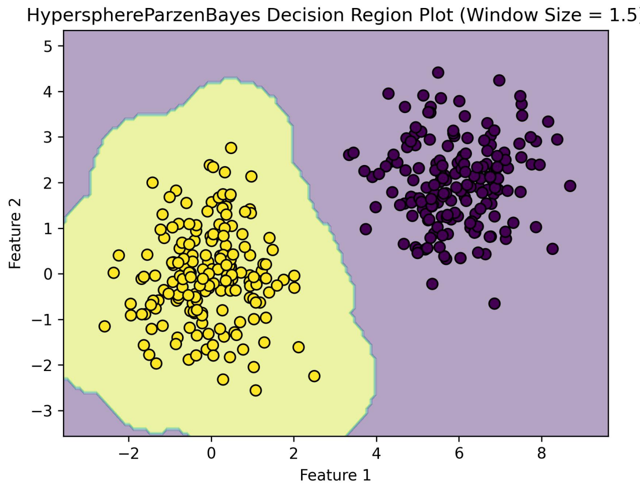
1.4.1 Plots of the Decision Surface



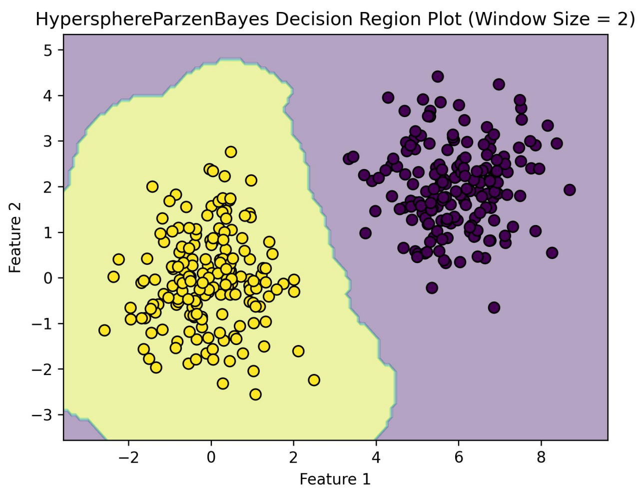
(1) $R = 0.01$



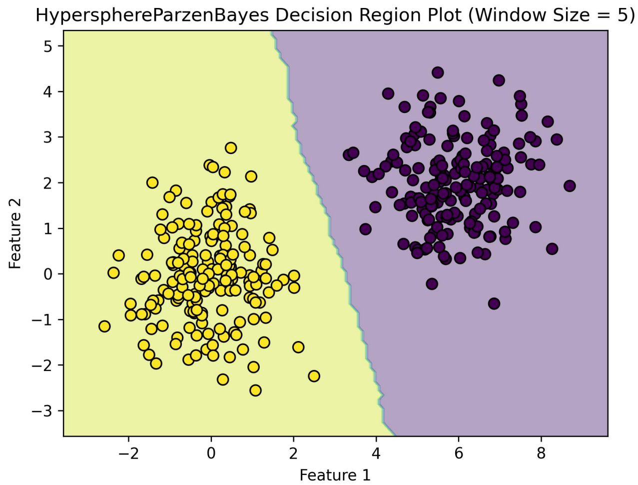
(2) $R = 0.1$


HypersphereParzenBayes Decision Region Plot (Window Size = 1.5)
Feature 2
Feature 1
5
4
3
2
1
0
−1
−2
−3
−2
0
2
4
6
8

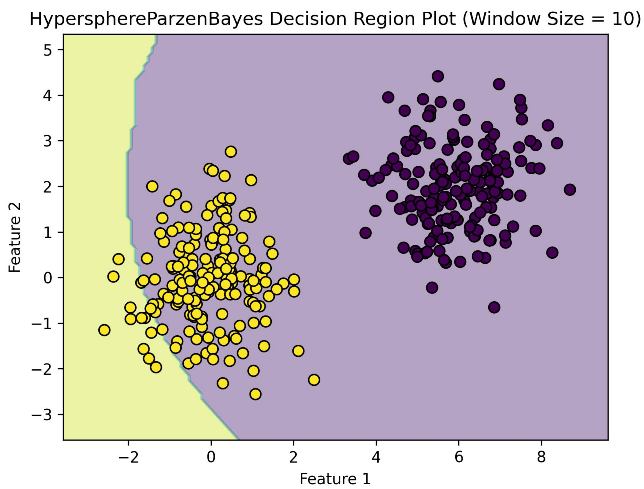
(3) $R = 1.5$


HypersphereParzenBayes Decision Region Plot (Window Size = 2)
Feature 2
Feature 1
5
4
3
2
1
0
−1
−2
−3
−2
0
2
4
6
8

(4) $R = 2$



(5) $R = 5$



(6) $R = 10$

Figure 4: Decision Surface of the Classifier for different R

1.4.2 Training, Validation and Testing Accuracy

R	Training Accuracy	Validation Accuracy	Testing Accuracy
0.01	1.0	0.522 22	0.42
0.1	1.0	0.666 67	0.66
0.5	1.0	0.977 78	0.98
1.0	1.0	1.0	1.0
1.5	1.0	1.0	1.0
2.0	**1.0**	**1.0**	**1.0**
5.0	1.0	1.0	1.0
10.0	0.541 67	1.0	0.48

Table 13: Training, Validation and Testing Accuracy for different K

1.4.3 Inferences

1. The application of the Parzen window method to the dataset provides insights into the model's performance, notably influenced by the parameter R.

2. For smaller R values (0.01, 0.1), the model exhibits lower accuracy, particularly noticeable in the testing dataset. This suggests challenges in capturing the intricate patterns with excessively small R.

3. At an optimal R of 1.0, the Parzen window method achieves perfect accuracy across all datasets, showcasing its ability to precisely capture the class-conditional probabilities.

4. Larger R values (5.0, 10.0) lead to a decrease in accuracy, especially noticeable in the training set for $R = 10.0$. This indicates that excessively large R values may result in underfitting.

5. The Parzen window method, when configured with an appropriate R, demonstrates effective class-conditional probability estimation on this dataset. The choice of R is critical, balancing model complexity and accuracy.

1.4.4 Confusion Matrix for the classifier

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	182	0
Class 1	0	178

Table 14: Confusion Matrix for training dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	47	0
Class 1	0	43

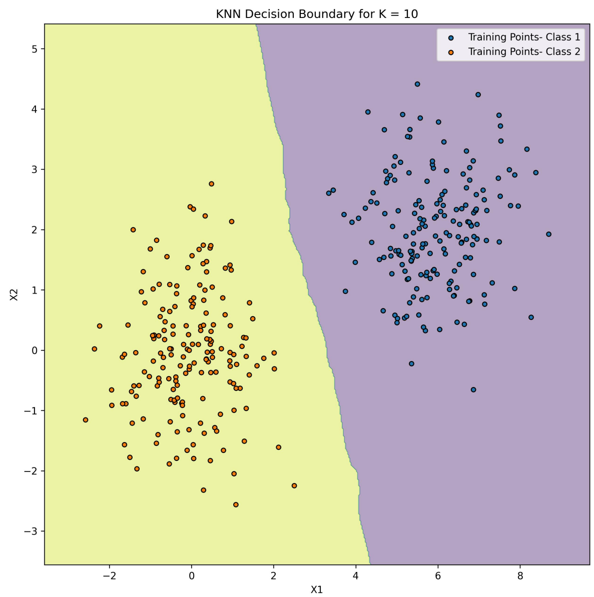
Table 15: Confusion Matrix for validation dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	21	0
Class 1	0	29

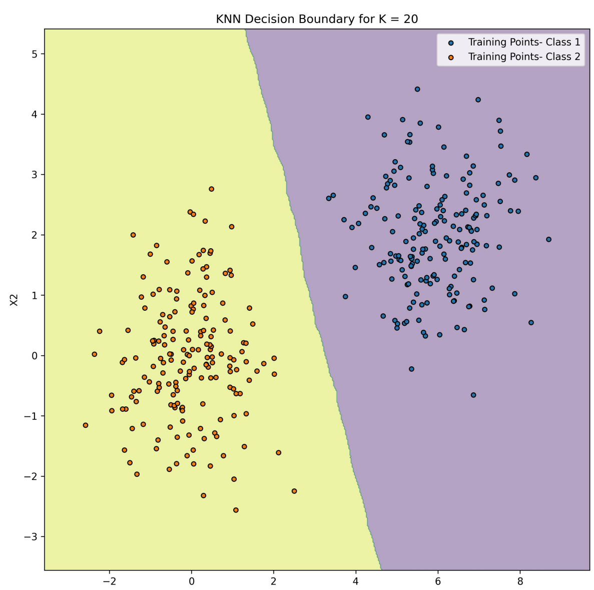
Table 16: Confusion Matrix for testing dataset

1.5 Bayes Classifier with K-Nearest Neighbours for estimating class-conditional probabilities

1.5.1 Plots of the Decision Surface



(1) $K = 10$



(2) $K = 20$

Figure 5: Decision Surface of the Classifier for different K

1.5.2 Hyperparameters v/s Training, Validation and Testing Accuracy

K	Training Accuracy	Validation Accuracy	Testing Accuracy
10	1.0	1.0	1.0
20	**1.0**	**1.0**	**1.0**

Table 17: Training, Validation and Testing Accuracy for different K

1.5.3 Inferences

1. From the distribution of the training data points, we find that the data is clearly linearly separable. Hence, K-Nearest Neighbours algorithm is well suited to calculate the class-conditional probabilities for this dataset.
2. This is reflected in the classification accuracy of the model developed. Even with large $K = 20$, the model performs exceptionally well scoring 100% accuracy in all of training, validation and testing datasets.
3. Since smaller values of K generally lead to an overfit with the training dataset, we take the model with $K = 20$ to the best model, as it is likely to avoid overfitting by taking more of the surrounding data into consideration for calculating the class-conditional probability.
4. However larger values of K tend to have a smoothing effect, due to which the model tends to underfit the data which leads to a more generalized nature of the decision boundary.
5. We also note that it is not possible to obtain the optimal value of K unless we have a larger dataset to test the model extensively.

1.5.4 Confusion Matrix for the best classifier

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	182	0
Class 1	0	178

Table 18: Confusion Matrix for training dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	47	0
Class 1	0	43

Table 19: Confusion Matrix for validation dataset

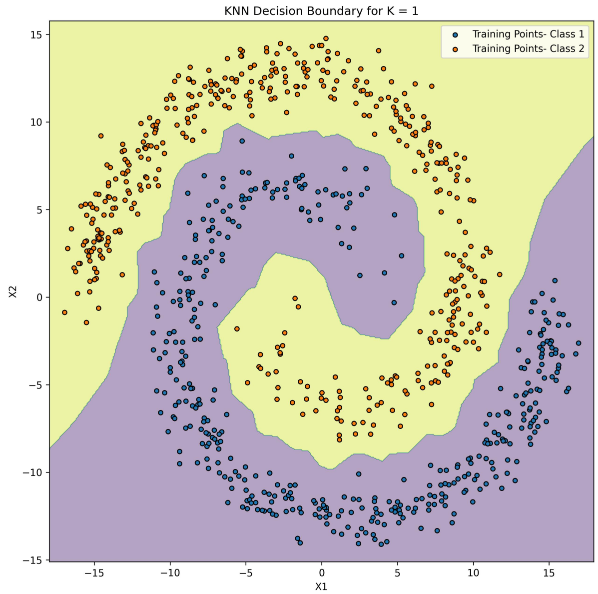
	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	21	0
Class 1	0	29

Table 20: Confusion Matrix for testing dataset

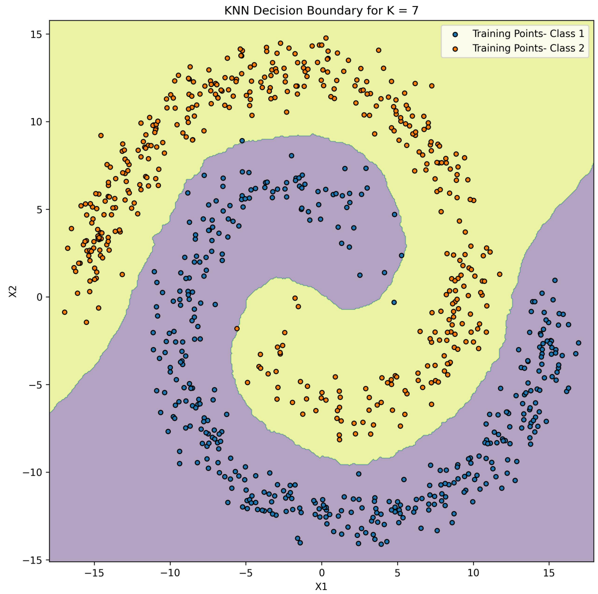
2 Dataset 1(b)

2.1 K-Nearest Neighbours Classifier

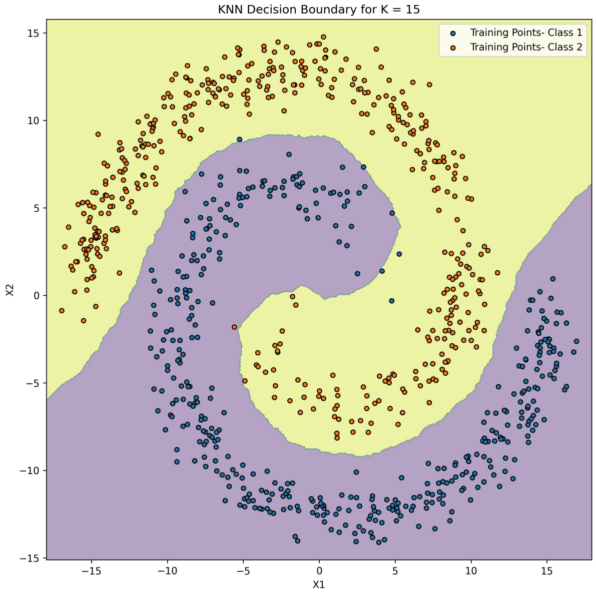
2.1.1 Plots of the Decision Surface



(1) $K = 1$



(2) $K = 7$



(3) $K = 15$

Figure 6: Decision Surface of the Classifier for different K

2.1.2 Hyperparameters v/s Training, Validation and Testing Accuracy

K	Training Accuracy	Validation Accuracy	Testing Accuracy
1	**1.0**	**1.0**	**1.0**
7	0.996 43	1.0	1.0
15	0.992 87	1.0	1.0

Table 21: Training, Validation and Testing Accuracy for different K

2.1.3 Inferences

1. The simple KNN classifier exhibits impeccable performance across all tested scenarios.
2. With $K = 1$, the model achieves perfect accuracy on the training, validation, and testing datasets, indicating a precise fit to the data.
3. As K increases to 7 and 15, the model maintains high accuracy, showcasing its robustness to variations in K.
4. Notably, the KNN classifier demonstrates a remarkable ability to generalize well, evidenced by consistent high accuracy across different K values.
5. The simplicity and effectiveness of the KNN model make it a compelling choice for this dataset, as it seamlessly adapts to varying complexities in the underlying patterns.

2.1.4 Confusion Matrix for the best classifier

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	182	0
Class 1	0	178

Table 22: Confusion Matrix for training dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	47	0
Class 1	0	43

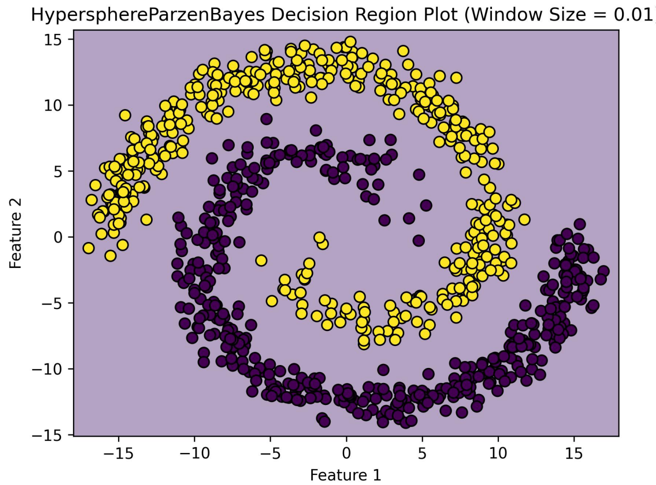
Table 23: Confusion Matrix for validation dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	21	0
Class 1	0	29

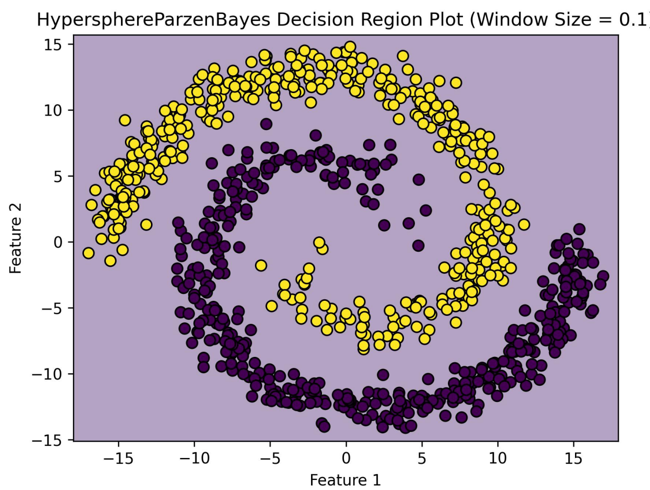
Table 24: Confusion Matrix for testing dataset

2.2 Bayes Classifier with Hypersphere-based Parzen-window density estimation

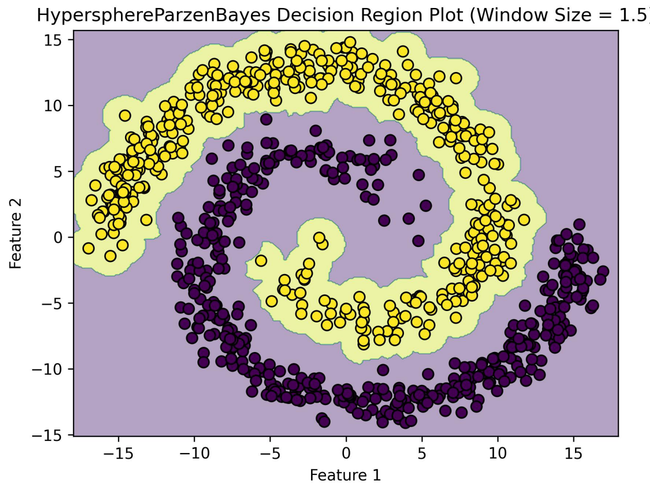
2.2.1 Plots of the decision surface



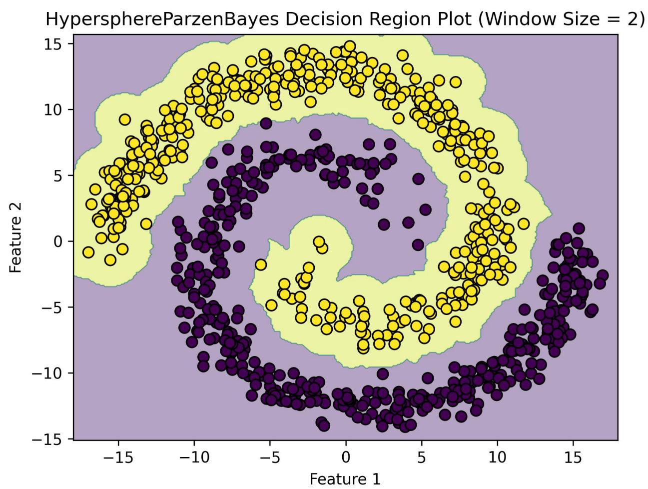
(1) $R = 0.01$



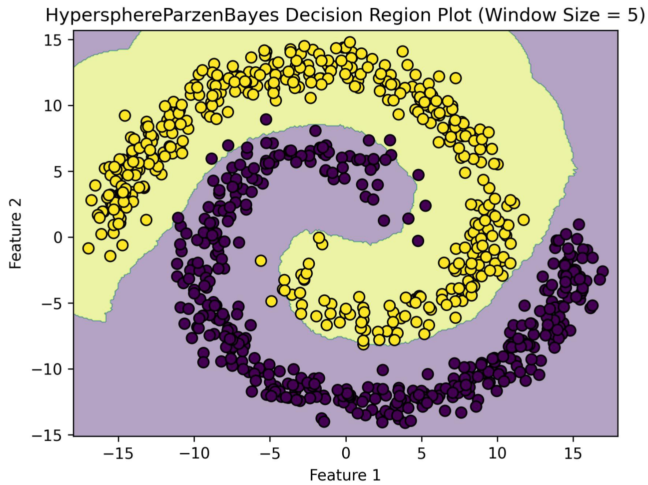
(2) $R = 0.1$


HypersphereParzenBayes Decision Region Plot (Window Size = 1.5)
Feature 1
Feature 2
15
10
5
0
−5
−10
−15
−15
−10
−5
0
5
10
15

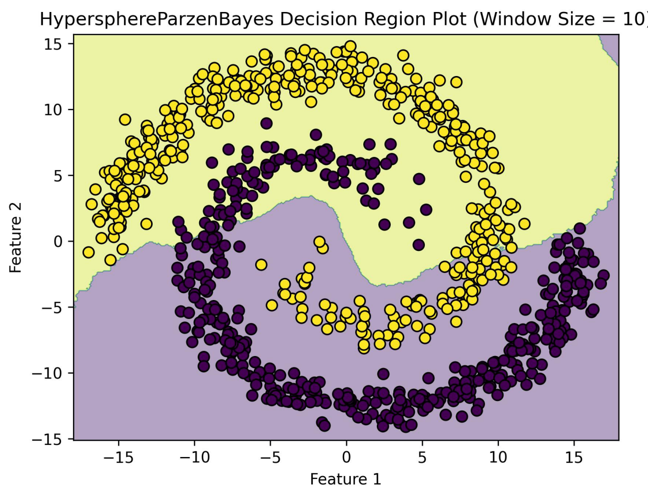
(3) $R = 1.5$


HypersphereParzenBayes Decision Region Plot (Window Size = 2)
Feature 1
Feature 2
15
10
5
0
−5
−10
−15
−15
−10
−5
0
5
10
15

(4) $R = 2$



(5) $R = 5$



(6) $R = 10$

Figure 7: Decision Surface of the Classifier for different R

2.2.2 Training, Validation and Testing Accuracy

R	Training Accuracy	Validation Accuracy	Testing Accuracy
0.01	1.0	0.521 08	0.42
0.1	1.0	0.558 82	0.66
0.5	1.0	0.928 57	0.98
1.0	1.0	0.983 19	1.0
1.5	**1.0**	**0.99579**	**1.0**
2.0	0.998 81	0.995 79	1.0
5.0	0.986 94	0.962 18	1.0
10.0	0.781 47	0.735 29	0.48

Table 25: Training, Validation and Testing Accuracy for different R

2.2.3 Inferences

1. The Bayes Classifier with Hypersphere-based Parzen-window density estimation consistently achieves high training accuracy, often approaching 1.0, indicating a robust fit to the training data.

2. Validation accuracy varies with R, peaking at $R = 2.0$ and showing a slight dip for extreme values. Testing accuracy remains generally high, with a notable drop for $R = 10.0$.

3. Small R values (0.01, 0.1) result in lower validation and testing accuracies, suggesting challenges in capturing underlying patterns with overly small R.

4. Optimal performance is observed around $R = 1.0$, 1.5, and 2.0, indicating a balanced representation of the underlying density patterns.

5. Larger R values (5.0, 10.0) lead to decreased model performance, particularly noticeable for $R = 10.0$. This suggests that excessively large R values may cause over-smoothing and a less accurate representation of the data distribution.

6. The model's sensitivity to the choice of R underscores the importance of tuning this parameter judiciously to achieve optimal performance.

2.2.4 Confusion Matrix for the best classifier

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	182	0
Class 1	0	178

Table 26: Confusion Matrix for training dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	47	0
Class 1	0	43

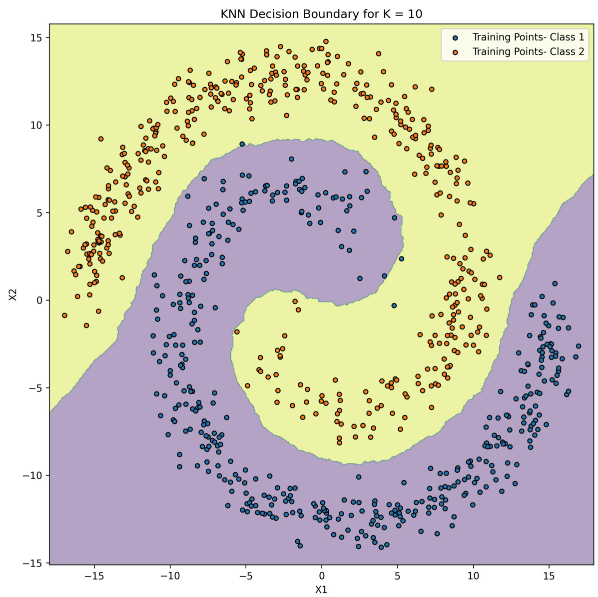
Table 27: Confusion Matrix for validation dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	21	0
Class 1	0	29

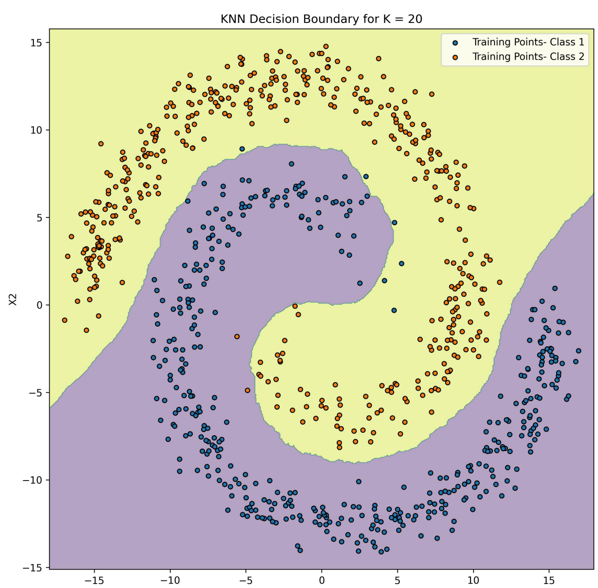
Table 28: Confusion Matrix for testing dataset

2.3 Bayes Classifier with K-Nearest Neighbours for estimating class-conditional probabilities

2.3.1 Plots of the Decision Surface



(1) $K = 10$



(2) $K = 20$

Figure 8: Decision Surface of the Classifier for different K

2.3.2 Hyperparameters v/s Training, Validation and Testing Accuracy

K	Training Accuracy	Validation Accuracy	Testing Accuracy
10	**0.99644**	**1.0**	**1.0**
20	0.990 49	0.987 39	1.0

Table 29: Training, Validation and Testing Accuracy for different K.

2.3.3 Inferences

1. From the distribution of the training data points, we can see the data is not linearly separable. To use K-Nearest Neighbours algorithm to calculate the class-conditional probability requires to carefully choose the value of K.

2. The models performs exceptionally well with both $K = 10, 20$. This indicates that there is a clear separation between the classes.

3. $K = 20$ gives a slightly inferior performance on both training and validation datasets. This indicates that the separation between the classes might not be large enough to accommodate such a large number of neighbours and hence has led to an underfitting scenario.

4. Given that $K = 10$ gives a better performance as compared to $K = 20$ we can also conclude that the closer data points exert more influence of the nature of a sample point than those points farther away.

2.3.4 Confusion Matrix for the best classifier

	Predicted Class	
Predicted Class	**Class 0**	**Class 1**
Class 0	408	3
Class 1	0	431

Table 30: Confusion Matrix for training dataset

	Predicted Class	
Predicted Class	**Class 0**	**Class 1**
Class 0	124	0
Class 1	0	114

Table 31: Confusion Matrix for validation dataset

	Predicted Class	
Predicted Class	**Class 0**	**Class 1**
Class 0	62	0
Class 1	0	58

Table 32: Confusion Matrix for testing dataset

2.4 Bayes Classifier using a Gaussian Distribution with full covariance matrices

2.4.1 Plot of Decision Surface and Level Curves

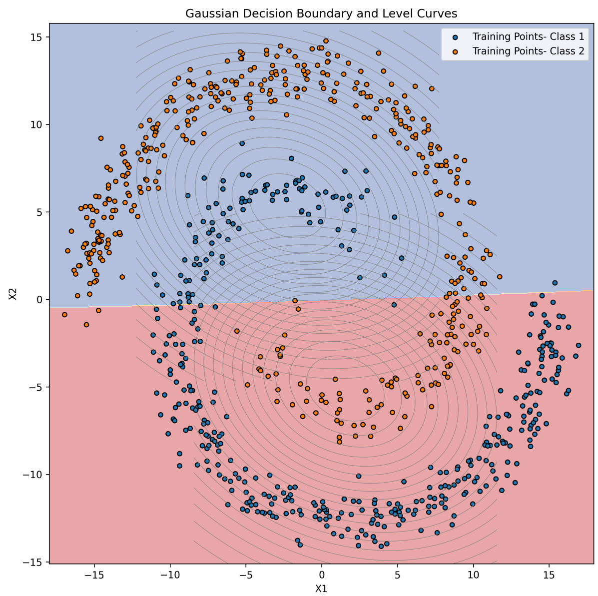


Figure 9: Decision Surface of the Classifier and Level Curves of the Gaussian Distribution

2.4.2 Training, Validation and Testing Accuracy

Training Accuracy	Validation Accuracy	Testing Accuracy
0.755 34	0.726 89	0.708 33

Table 33: Training, Validation and Testing Accuracy of the classifier.

2.4.3 Inferences

1. From the distribution of the data points, we can infer that it is not possible to estimate the entire set of points with a single Gaussian distribution. Hence, we expect that the model does not perform well in this scenario.
2. As expected, the model indeed, highly underfits the data and generates a very generalized decision surface.
3. The poor performance of the model also shows that the decision surface between the classes exhibits a high degree of complexity. A single Gaussian distribution is inadequate to represent such a complex boundary.
4. A more complex model leveraging better estimation methods such as Gaussian Mixture Models are likely to perform better in this situation.

2.4.4 Confusion Matrix of the classifier

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	312	99
Class 1	107	324

Table 34: Confusion Matrix for training dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	88	36
Class 1	29	85

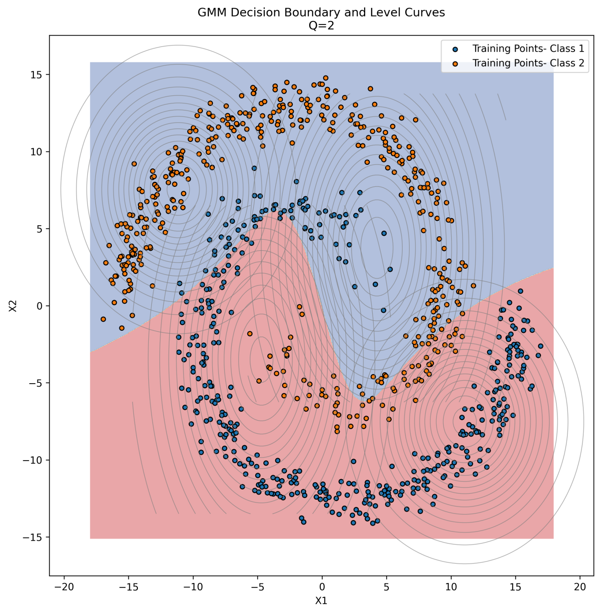
Table 35: Confusion Matrix for validation dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	45	17
Class 1	18	4

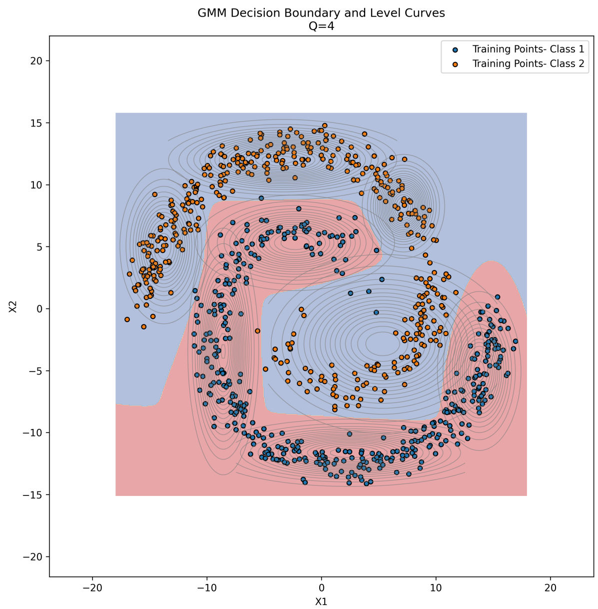
Table 36: Confusion Matrix for testing dataset

2.5 Bayes Classifier using Gaussian Mixture Model with diagonal covariance matrices

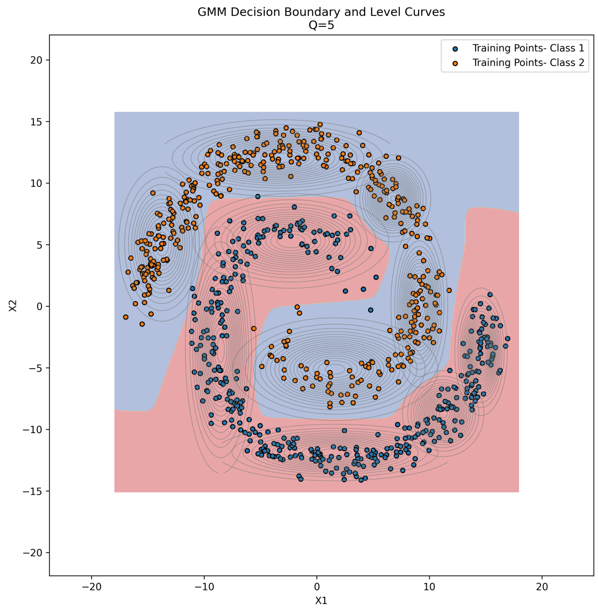
2.5.1 Plots of Decision Surface and Level Curves



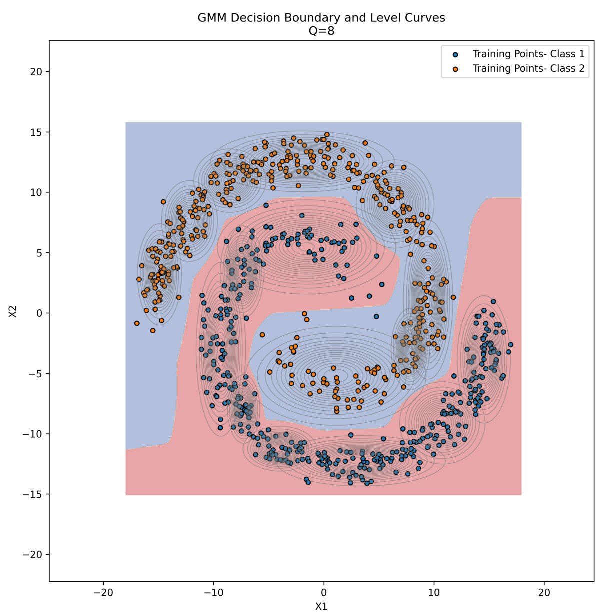
(1) $Q = 2$



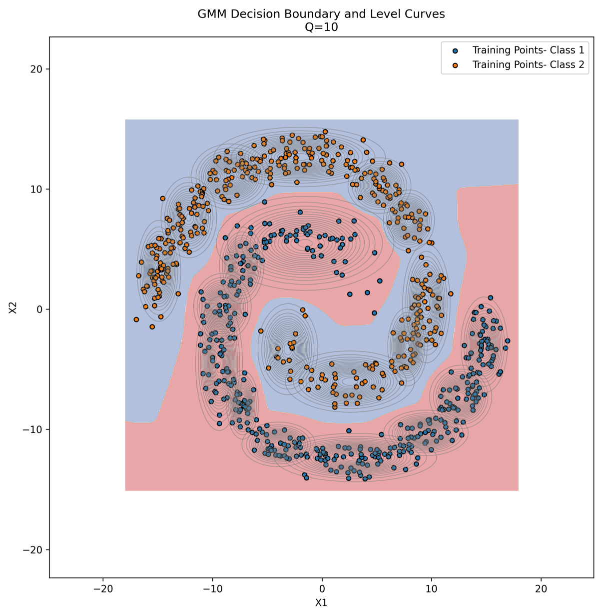
(2) $Q = 4$


GMM Decision Boundary and Level Curves
Q=5
Training Points- Class 1
Training Points- Class 2
X1
X2

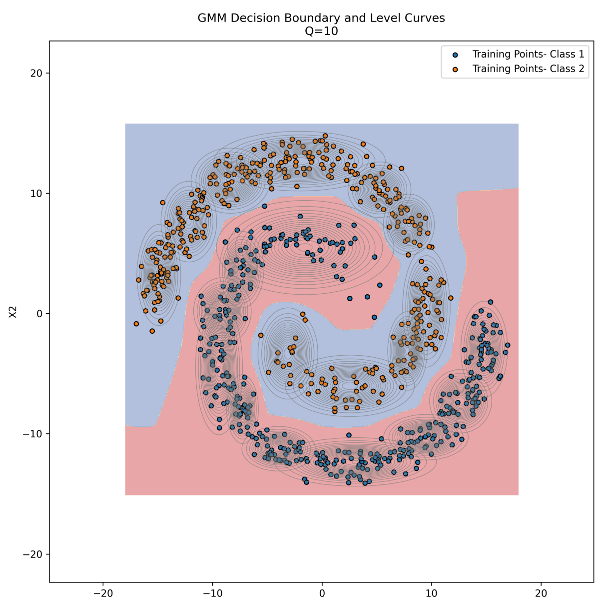
(3) $Q = 5$


GMM Decision Boundary and Level Curves
Q=8
Training Points- Class 1
Training Points- Class 2
X1
X2

(4) $Q = 8$



(5) $Q = 10$



(6) $Q = 12$

Figure 10: Decision Surface of Classifier and Level Curves of the Gaussian Components

2.5.2 Hyperparameters v/s Training, Validation and Testing Accuracy

Q	Training Accuracy	Validation Accuracy	Testing Accuracy
2	0.827 79	0.777 31	0.825
4	0.991 68	0.970 58	1.0
5	0.995 24	0.991 59	1.0
8	0.998 81	0.995 79	1.0
10	1.0	0.995 79	1.0
12	**1.0**	**1.0**	**1.0**

Table 37: Training, Validation and Testing Accuracy for different Q.

2.5.3 Inferences

1. From the distribution of the training data points, we see that the classes are not linearly separable. GMMs will be able to give a good estimate of the decision boundary given that the number of components are chosen carefully.
2. For small values of number of components, $Q = 2, 4, 5$, we see that the model is unable to estimate the decision boundary accurately. This shows that the decision boundary exhibits a higher degree of probability.
3. The usage of diagonal covariance matrices also reduces the performance of the model as the complexity of the model is reduced.
4. As the number of components, Q, increases, we find the model becomes better at approximating the best decision boundary between the two classes.
5. $Q = 12$ achieves the best performance, scoring 100% accuracy on all of training, validation and testing datasets.
6. Increasing Q further might lead to overfitting on the training dataset which will lead to poor performance on the validation and testing datasets.

2.5.4 Confusion Matrix of the best classifier

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	411	0
Class 1	0	431

Table 38: Confusion Matrix for training dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	124	0
Class 1	0	114

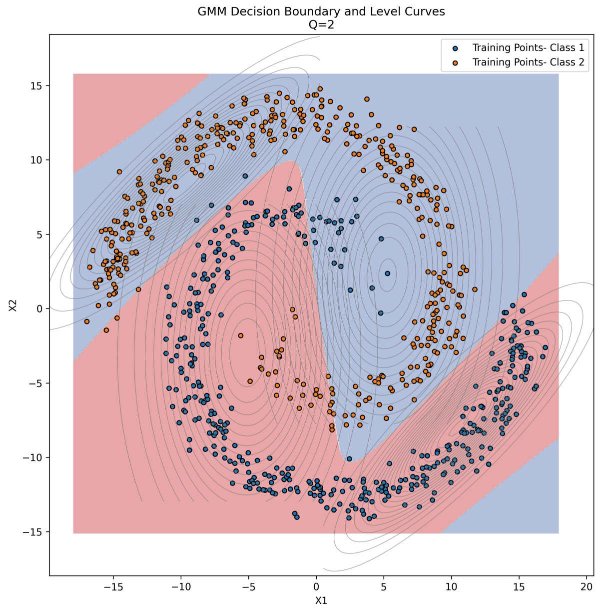
Table 39: Confusion Matrix for validation dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	62	0
Class 1	0	58

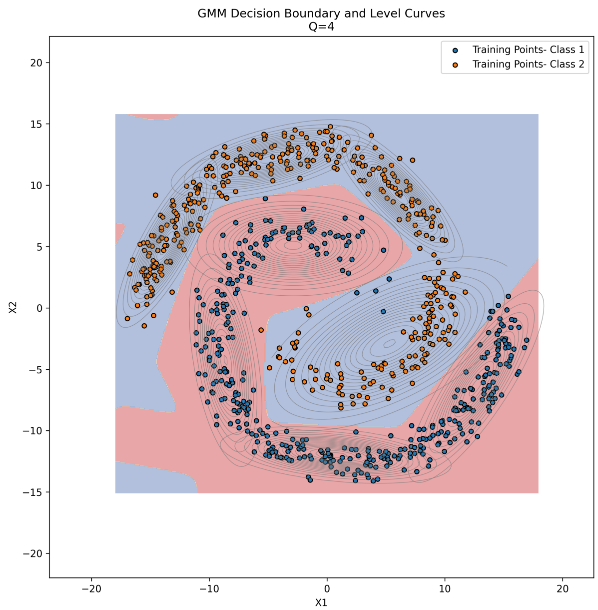
Table 40: Confusion Matrix for testing dataset

2.6 Bayes Classifier using Gaussian Mixture Model with full covariance matrices

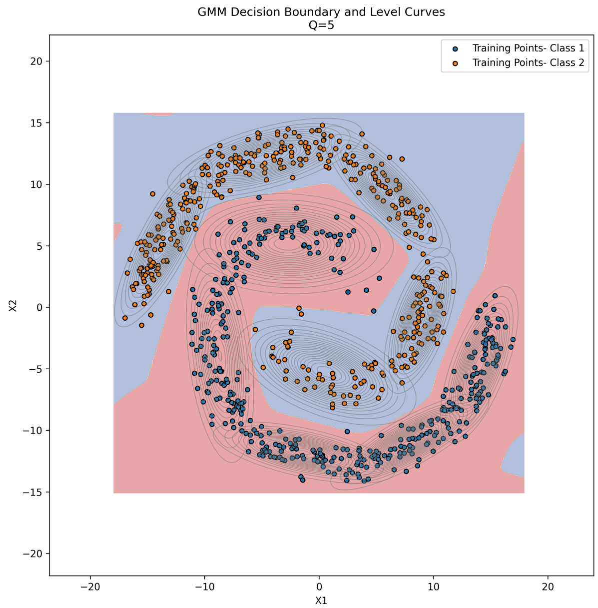
2.6.1 Plots of Decision Surface and Level Curves



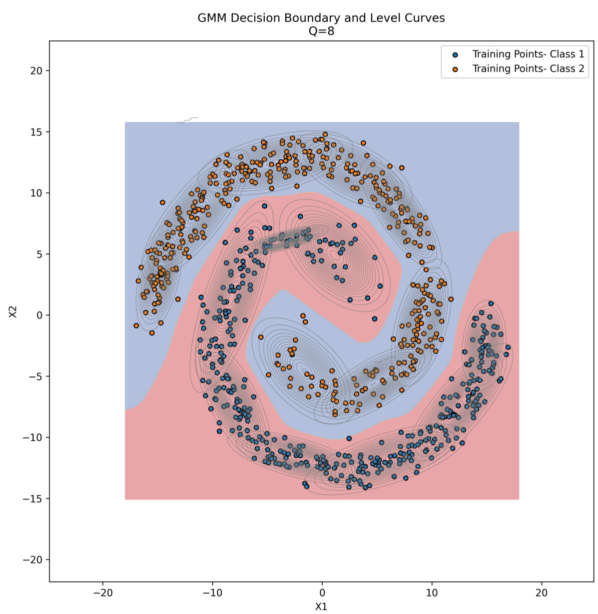
(1) $Q = 2$



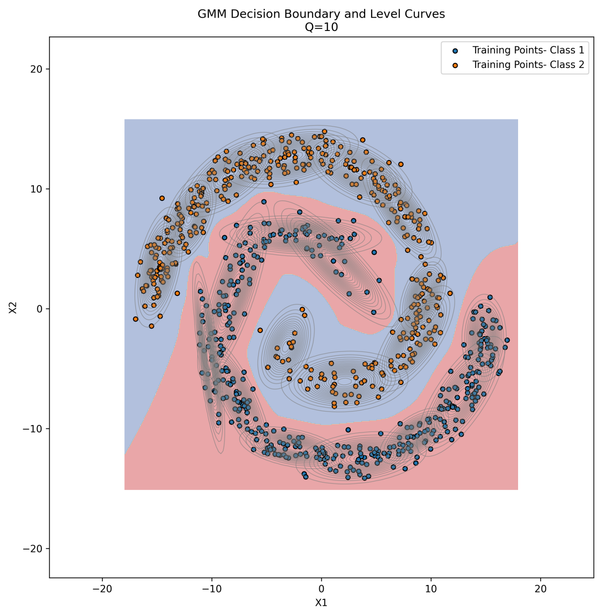
(2) $Q = 4$


GMM Decision Boundary and Level Curves
Q=5
Training Points- Class 1
Training Points- Class 2
X1
X2

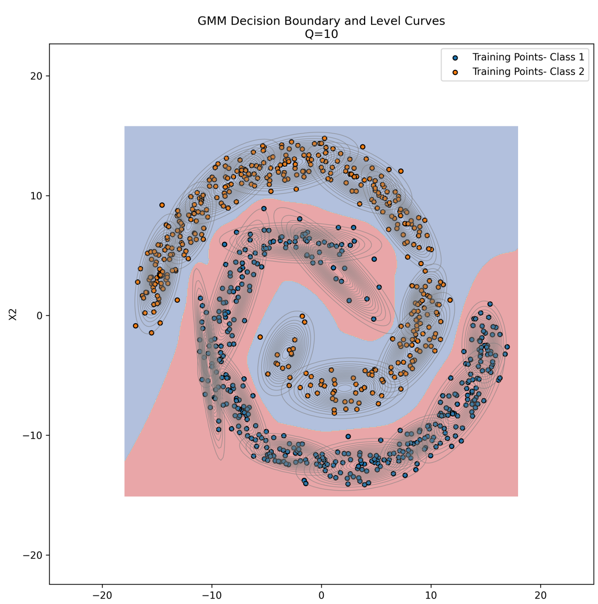
(3) $Q = 5$


GMM Decision Boundary and Level Curves
Q=8
Training Points- Class 1
Training Points- Class 2
X1
X2

(4) $Q = 8$



(5) $Q = 10$



(6) $Q = 12$

Figure 11: Decision Surface of Classifier and Level Curves of the Gaussian Components

2.6.2 Hyperparameters v/s Training, Validation and Testing Accuracy

Q	Training Accuracy	Validation Accuracy	Testing Accuracy
2	0.928 74	0.882 35	0.916 66
4	0.990 49	0.987 39	1.0
5	1.0	0.995 79	1.0
8	1.0	0.995 79	1.0
10	**1.0**	**1.0**	**1.0**
12	1.0	0.991 59	1.0

Table 41: Training, Validation and Testing Accuracy for different Q.

2.6.3 Inferences

1. In comparison to the model using only diagonal covariance matrices, this model has better performance even for small values of Q, as the complexity of the model is increased by using full covariance matrices.
2. Even so, with smaller $Q = 2, 4$, the model does not completely capture the nature of the distribution.
3. As Q increases, the model performance improves on the validation dataset, meaning that the points local to the point exert more influence than those farther away.
4. However, for large $Q = 12$, the model begins to overfit on the training data and the performance on the validation drops.
5. The best model here is $Q = 10$ which achieves 100% accuracy on all of training, validation and testing datasets.

2.6.4 Confusion Matrix of the best classifier

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	411	0
Class 1	0	431

Table 42: Confusion Matrix for training dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	124	0
Class 1	0	114

Table 43: Confusion Matrix for validation dataset

	Predicted Class	
Actual Class	**Class 0**	**Class 1**
Class 0	62	0
Class 1	0	58

Table 44: Confusion Matrix for testing dataset

3 Dataset 2

3.1 Inferences about the dataset

1. The given dataset corresponds to image data and hence we expect it to be highly complex.
2. Simple models such as KNNs and Bayes Classifiers are unlikely to provide good performance on such data owing to the complex nature of the data.
3. Such models are generally unable to estimate highly complex decision boundaries.
4. Good classification accuracy can be obtained by leveraging more complex models such as Multi-Layer Feed-Forward Neural Networks and Convolutional Neural Networks.

3.2 K-Nearest Neighbours Classifier

3.2.1 Hyperparameters v/s Training, Validation and Testing Accuracy

K	Training Accuracy	Validation Accuracy	Testing Accuracy
1	1	0.534	0.466
7	**0.63228**	**0.573**	**0.542**
15	0.597 42	0.559	0.55

Table 45: Training, Validation, and Testing Accuracy for different K.

3.2.2 Inferences

1. Applying the KNN classifier to the dataset reveals insights into its performance, which is notably influenced by the choice of K.
2. With $K = 1$, the model achieves perfect accuracy on the training set, yet exhibits a substantial drop in accuracy on both validation and testing datasets. This suggests overfitting to the training data and lack of generalization.
3. As K increases to 7 and 15, the model's accuracy on the training set decreases, indicating a move away from overfitting. However, the validation and testing accuracies improve, suggesting a better balance between model complexity and generalization.
4. The observed trend emphasizes the importance of selecting an appropriate K to strike a balance between capturing local patterns and achieving a more generalized model.
5. The KNN classifier's performance, with careful consideration of K, can offer a balance between model complexity and generalization on this dataset.

3.2.3 Confusion Matrix of the best classifier

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	337	141	24	38	160
Class 1	13	556	36	69	26
Class 2	1	245	352	89	13
Class 3	4	187	56	440	13
Class 4	46	85	14	27	528

Table 46: Confusion Matrix for the training dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	71	42	6	9	72
Class 1	5	158	10	18	9
Class 2	0	71	92	30	7
Class 3	2	60	24	111	3
Class 4	15	32	4	8	141

Table 47: Confusion Matrix for the validation dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	34	21	6	6	33
Class 1	1	73	8	13	5
Class 2	1	27	45	25	2
Class 3	1	36	14	49	0
Class 4	8	13	2	7	70

Table 48: Confusion Matrix for the testing dataset

3.3 Bayes Classifier with Hypersphere-based Parzen-window density estimation

3.3.1 Hyperparameters v/s Training, Validation and Testing Accuracy

Window Size	Training Accuracy	Validation Accuracy	Test Accuracy
0.01	1	0.2	0.2
0.1	1	0.2	0.2
0.5	0.971 14	0.25	0.24
1	**0.35285**	**0.349**	**0.342**
1.5	0.2	0.2	0.2
2	0.2	0.2	0.2
5	0.2	0.2	0.2
10	0.2	0.2	0.2

Table 49: Window Size and Accuracy Metrics

3.3.2 Inferences

1. Given the complex nature of the data, the application of the Bayes Classifier with Hypersphere-based Parzen-window density estimation indicates challenges. The choice of window size significantly impacts model performance.

2. Smaller window sizes (0.01, 0.1) result in very low accuracy across all datasets, suggesting a struggle to capture meaningful information about the underlying distribution.

3. At a window size of 0.5, the model shows improved accuracy, indicating a better grasp of the data's patterns. However, the performance drops substantially for larger window sizes (1, 1.5, 2, 5, 10).

4. The observed trend suggests that an optimal window size balancing information capture and model complexity is crucial for achieving meaningful results.

5. The Bayes Classifier with Hypersphere-based Parzen-window density estimation, when carefully configured, can provide insights into the complex nature of the data. However, the choice of window size requires meticulous consideration.

3.3.3 Confusion Matrix of the best classifier

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	38	342	12	75	233
Class 1	0	383	92	205	20
Class 2	0	201	244	249	6
Class 3	0	315	112	265	8
Class 4	3	305	8	79	305

Table 50: Confusion Matrix for the training dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	11	89	8	14	78
Class 1	0	106	32	56	6
Class 2	0	62	70	67	1
Class 3	0	85	40	75	0
Class 4	0	89	3	21	87

Table 51: Confusion Matrix for the validation dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	3	47	2	9	39
Class 1	0	49	12	35	4
Class 2	0	27	37	34	2
Class 3	0	43	17	40	0
Class 4	0	46	2	10	42

Table 52: Confusion Matrix for the testing dataset

3.4 Bayes Classifier using K-Nearest Neighbours for estimation class-conditional probabilities

3.4.1 Hyperparameters v/s Training, Validation and Testing Accuracy

K	Training Accuracy	Validation Accuracy	Testing Accuracy
10	0.61	0.553	0.52
20	**0.59257**	**0.564**	**0.528**

Table 53: Training, Validation and Testing Accuracy for different K.

3.4.2 Inferences

1. Given the complex nature of the data, using K-Nearest Neighbours for calculating the class-conditional probability is likely to be highly inefficient.
2. As expected the model, captures very less information regarding the underlying distribution of data.
3. The performance of the model is only slightly better than a random classifier which has a performance of 50%.

3.4.3 Confusion Matrix of the best classifier

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	284	172	22	38	184
Class 1	6	560	35	66	33
Class 2	1	279	324	85	11
Class 3	5	238	53	391	13
Class 4	36	113	17	19	515

Table 54: Confusion Matrix for training dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	68	41	6	9	76
Class 1	2	162	9	15	12
Class 2	0	83	96	16	5
Class 3	0	72	23	101	4
Class 4	16	36	2	9	137

Table 55: Confusion Matrix for validation dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	33	21	5	8	33
Class 1	1	76	6	10	7
Class 2	1	34	41	22	2
Class 3	1	38	14	47	0
Class 4	10	18	0	5	67

Table 56: Confusion Matrix for testing dataset

3.5 Bayes Classifier using a Gaussian Distribution with full covariance matrices

3.5.1 Training, Validation and Testing Accuracy

Training Accuracy	Validation Accuracy	Testing Accuracy
0.865 14	0.557	0.55

Table 57: Training, Validation and Testing Accuracy of the classifier

3.5.2 Inferences

1. This model is once again not sufficient to capture the underlying distribution of the data points.
2. A single Gaussian distribution is not sufficient to capture the complex nature of the data.
3. The model performs better than the model using KNN algorithm for calculating the class-conditional probability on the training dataset, but not on the validation dataset. This could mean that the model is overfitting on the data.

3.5.3 Confusion Matrix of the classifier

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	550	17	19	22	87
Class 1	17	608	33	20	22
Class 2	13	38	609	25	15
Class 3	9	31	23	625	12
Class 4	27	16	15	11	631

Table 58: Confusion Matrix for training dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	90	23	10	10	67
Class 1	29	90	42	27	12
Class 2	7	26	130	31	6
Class 3	8	36	44	101	11
Class 4	25	17	9	3	146

Table 59: Confusion Matrix for validation dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	52	4	10	7	27
Class 1	5	49	21	18	7
Class 2	4	9	61	21	5
Class 3	0	23	23	49	5
Class 4	17	8	3	8	64

Table 60: Confusion Matrix for testing dataset

3.6 Bayes Classifier using Gaussian Mixture Model with diagonal covariance matrices

3.6.1 Hyperparameters v/s Training, Validation and Testing Accuracy

Q	Training Accuracy	Validation Accuracy	Testing Accuracy
2	0.589 43	0.552	0.558
4	**0.62771**	**0.573**	**0.52**
5	0.640 57	0.538	0.51
8	0.712 57	0.535	0.534
10	0.743 43	0.564	0.506
12	0.760 86	0.547	0.516

Table 61: Training, Validation, and Testing Accuracy for different Q.

3.6.2 Inferences

1. The usage of GMMs for calculating the class-conditional probability does not improve the performance of the Bayes Classifier. This indicates that the underlying data doesn't exhibit multi-modal nature. Hence, the GMM model is unable to capture the underlying distribution effectively.

2. As the number of components, Q, increases the model begins to overfit on the training data. This can be seen by the reducing validation accuracy as Q increases.

3. The usage of diagonal covariance matrices also reduces the complexity of the model. This can be seen by comparing the accuracy of the model against the model that leverages full covariance matrices. We can see that the corresponding model for each value of Q, has a slightly better accuracy.

3.6.3 Confusion Matrix of the best classifier

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	382	73	47	41	157
Class 1	33	435	96	95	41
Class 2	16	98	468	96	22
Class 3	11	119	113	432	25
Class 4	114	55	25	26	480

Table 62: Confusion Matrix for training dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	91	27	10	10	62
Class 1	10	110	35	31	14
Class 2	1	29	132	32	6
Class 3	1	41	41	107	10
Class 4	36	17	6	8	133

Table 63: Confusion Matrix for validation dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	47	11	8	5	29
Class 1	4	58	18	15	5
Class 2	2	25	41	28	4
Class 3	1	21	24	54	0
Class 4	18	11	6	5	60

Table 64: Confusion Matrix for testing dataset

3.7 Bayes Classifier using Gaussian Mixture Model with full covariance matrices

3.7.1 Hyperparameters v/s Training, Validation and Testing Accuracy

Q	Training Accuracy	Validation Accuracy	Testing Accuracy
2	0.591 43	0.548	0.56
4	**0.63229**	**0.558**	**0.476**
5	0.645 14	0.538	0.518
8	0.705 14	0.55	0.528
10	0.739 43	0.554	0.536
12	0.766 29	0.531	0.52

Table 65: Training, Validation, and Testing Accuracy for different Q.

3.7.2 Inferences

1. Even with the usage of full covariance matrices, the model is unable to improve performance on this dataset. This owes to the fact that the underlying data is not multi-model and hence the GMM model is unable to capture the features of the distribution effectively.

2. As with the usage of diagonal covariance matrices, with the increase in Q, the model begins to overfit the data and this can be seen in the decreasing validation accuracy but increasing training accuracy. The model is learning the local features of the training dataset very well and hence is unable to generalize to the data.

3. However, with the usage of full covariance matrices, the complexity of the model increases. This slightly improves the performance of the model. As seen in the above table, for each Q, the model using full covariance matrices performs better than the model using only diagonal covariance matrices.

3.7.3 Confusion Matrix of the best classifier

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	409	73	27	38	153
Class 1	38	435	84	101	42
Class 2	15	107	447	97	34
Class 3	23	117	90	443	27
Class 4	125	45	23	28	479

Table 66: Confusion Matrix for training dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	100	24	11	5	60
Class 1	13	107	32	35	13
Class 2	2	39	117	32	10
Class 3	5	45	38	103	9
Class 4	36	18	10	5	131

Table 67: Confusion Matrix for validation dataset

	Predicted Class				
Actual Class	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 0	46	9	8	4	33
Class 1	4	47	26	15	8
Class 2	5	16	44	31	4
Class 3	4	22	24	48	2
Class 4	25	13	4	5	53

Table 68: Confusion Matrix for testing dataset