Implementation and Evaluation of Bayesian Classifiers Using Gaussian Distributions

Niti Priya ,*Research Intern , Indian Institute of Technology, Dharwad*

***Abstract*— The report is titled *Implementation and Evaluation of Bayesian Classifiers Using Gaussian Distributions*, exploring the design and performance of Bayesian classifiers. The Bayes' theorem underpinning these classifiers facilitates an estimate of posterior probabilities in an effort to gain the most optimal decision without maximum error rates. It implements Gaussian distributions for robust models for both univariate and multivariate data. Key topics include the roles of priors, likelihoods, and evidence in classification and the effects of covariance structures on decision boundaries. Through detailed analysis and evaluation, this study underscores the practical utility of Bayesian classifiers in diverse applications, which shows their adaptability and precision in handling real-world data.**

I. INTRODUCTION

The Bayesian classifier is one of the basic tools in statistical learning and pattern recognition, using Bayes theorem to classify data according to posterior probabilities. With prior knowledge and observed data, it gives a probabilistic approach to decision-making, minimizing error rates in classification tasks. This report discusses the implementation and evaluation of Bayesian classifiers, with special emphasis on the use of *Gaussian distributions* to model class data. Gaussian distributions are particularly useful because they are simple and efficient in modeling both univariate and multivariate data, making them ideal for this study.

In this work, each class is represented by its *mean vector* and *covariance matrix*, which capture the central tendency and spread of the data. The covariance matrices are especially crucial, as they provide information about the relationships between features, shaping the geometry of decision boundaries. This varies on how the covariance structure could go - linear or nonlinear in its boundaries. There can be *decision boundary plots* and *contour plots* to have this intuitive view of the how classifier discriminates classes.

The performance of the Bayesian classifier is tested with an amalgamation of metrics that include *accuracy*, *precision*, *recall*, and the *F-score*. Furthermore, the classifier's predictions are analyzed using a *confusion matrix*, which details true positives, true negatives, false positives, and false negatives. From these evaluations, this report portrays the effectiveness, robustness, and precision of the Bayesian classifier in real-world classification problems.

II. DATASET DESCRIPTION

A. Linearly Separable Dataset

This dataset consists of points that can be perfectly separated by a straight line or hyperplane in the feature space. It serves as an ideal starting point for testing the performance of models on well-defined, simple boundaries. The dataset's simplicity allows for a clear understanding of how effectively the models classify data with linear decision boundaries.

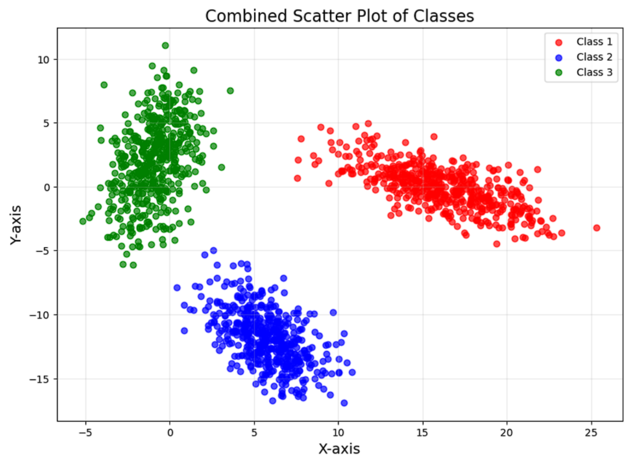

Fig. 1. Linearly Separable Dataset

B. Non Linearly Separable Dataset

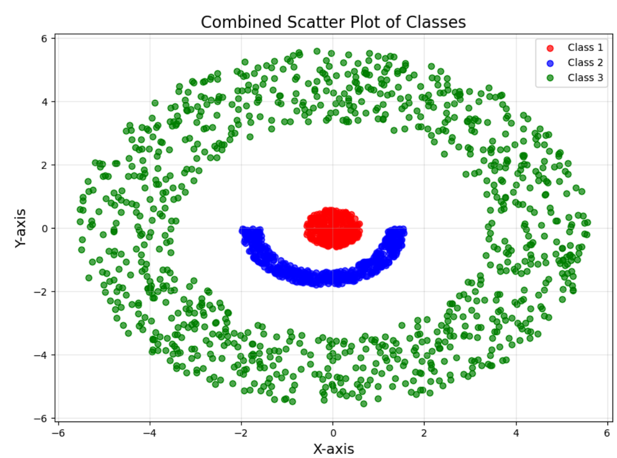

Fig. 2. Non-Linearly Separable Dataset

The second dataset is more complex, where no single straight line can separate the points. The decision boundaries are curved or intricate, and models must adapt to capture the data's non-linear relationships. This dataset tests how well models generalize beyond basic linear separability.

C. Real World Dataset

[The final dataset is for real-world scenarios, incorpo-

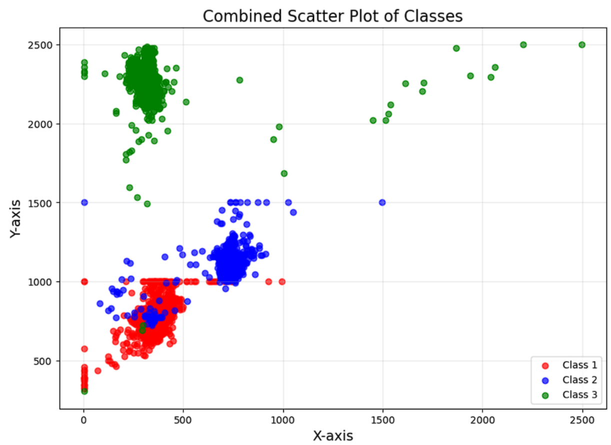

Fig. 3. Real-World Dataset

rating noisy and outlier data. Real-world data often contains irregularities, overlaps between classes, and inconsistent patterns, unlike artificially generated data. This dataset was used to evaluate how well the models handle imperfections and the complexity typical of real-world applications.

III. Methodology

1) **Data Preparation:** One must prepare or acquire a multi-class and multi-featured data set. For the demonstration, one can build synthetic datasets based on multivariate Gaussian distributions with predefined means and covariance matrices. Split the given dataset into a training set and test set to measure model scores effectively.
2) **Class-wise Statistical Parameter Estimation:** To present the center of the feature distribution, it's necessary to calculate the mean vector for each class from the training data.One should estimate the covariance matrix for every class in order to capture class-characteristic variability and correlation among features. In the case of diagonal covariance, only variances of individual features are preserved.

   a) ***Mean Vector*** *(μ_c):* For each class c, the mean vector of the feature data is calculated as:

$$\mu_c = \frac{1}{N_c} \sum_{i=1}^{N_c} X_i$$

   where:
   - N_c: Number of data points in class c
   - X_i: Feature vector of the i-th sample in class c

   b) ***Covariance Matrix*** *(Σ_c):* The covariance matrix for each class is computed as:

$$\Sigma_c = \frac{1}{N_c} \sum_{i=1}^{N_c} (X_i - \mu_c)(X_i - \mu_c)^T$$

   Diagonal covariance can be used by retaining only the diagonal elements of Σ_c.

   c) ***Prior Probability*** *($P(C = c)$):* The prior probability for each class c is estimated as:

$$P(C = c) = \frac{N_c}{N}$$

   where N is the total number of data points.
3) **Likelihood Calculation Using Gaussian Distribution:** The likelihood of a feature vector X given class c follows a multivariate Gaussian distribution:

$$P(X|C = c) = \frac{1}{(2\pi)^{d/2}|\Sigma_c|^{1/2}} \exp\left(-\frac{1}{2}(X - \mu_c)^T \Sigma_c^{-1}(X - \mu_c)\right)$$

   where:
   - d: Dimension of the feature space
   - $|\Sigma_c|$: Determinant of the covariance matrix Σ_c
   - Σ_c^{-1}: Inverse of the covariance matrix Σ_c
4) **Bayes' Decision Rule:**
   Bayes' Theorem computes the posterior probability of a class c for a given X:

$$P(C = c|X) = \frac{P(X|C = c)P(C = c)}{\sum_k P(X|C = k)P(C = k)}$$

   The classifier assigns X to the class with the highest posterior probability:

$$C^* = \arg\max_c P(C = c|X)$$

5) **Discriminant Function**
   For classification, the logarithmic form of the discriminant function is preferred for computational stability:

$$g_c(X) = -\frac{1}{2}(X - \mu_c)^T \Sigma_c^{-1}(X - \mu_c) - \frac{1}{2}\log|\Sigma_c| + \log P(C = c)$$

   where:
   - $g_c(X)$: Discriminant score for class c
   - X: Feature vector

   The class with the maximum discriminant score is chosen:

$$C^* = \arg\max_c g_c(X)$$

6) **Impact of Covariance Matrix on Decision Boundaries in Classification Models**

   Case 1: Isotropic Covariance Matrix
   - **Structure**: No correlation between features, diagonal matrix with same variances along the diagonal.
   - **Covariance Matrix**:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

   - **Decision Boundary**: Linear.

   Case 2. Full Covariance Matrix with Identical Covariance Pairs but Different Variances
   - **Structure**: Features have different variances, but equal covariance between all pairs.
   - **Covariance Matrix**:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma \\ \sigma & \sigma_2^2 \end{bmatrix}$$

   - **Decision Boundary**: Linear.

Case 3. Arbitary Covariance Matrix (Correlated Features)

- **Structure**: Features are correlated with different covariance values between each pair.
- **Covariance Matrix**:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix}$$

- **Decision Boundary**: Non-linear (elliptical or quadratic).

IV. Results and Analysis Across Dataset Types

A. Linear Dataset

Linear datasets are characterized by linearly separable classes, where the covariance matrices are diagonal, indicating no correlation between features. This scenario allows for straightforward classification with linear decision boundaries.

1) *Covariance Analysis and Decision Boundary*: Covariance matrices are computed for each class, retaining only variances along each feature due to their diagonal nature. This simplicity in covariance structure results in linear decision boundaries, which are visualized as straight lines or hyperplanes effectively partitioning the feature space and highlighting the classifier's ability to correctly separate classes.
   - *Case 1: Isotropic Covariance Matrix and Decision boundary*

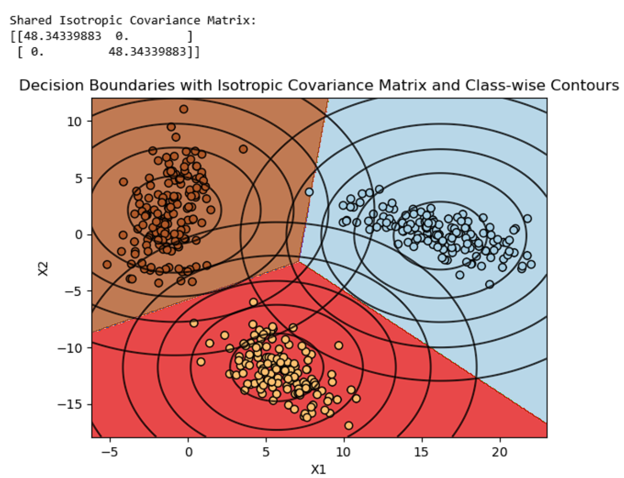

Fig. 4. Isotropic Covariance Matrix and Decision Boundary for Linearly Separable Data and Gaussian Class-wise Contours

- *Case 2: Full Covariance Matrix with identical Covariance Pairs but Different Variances and Decision Boundary*

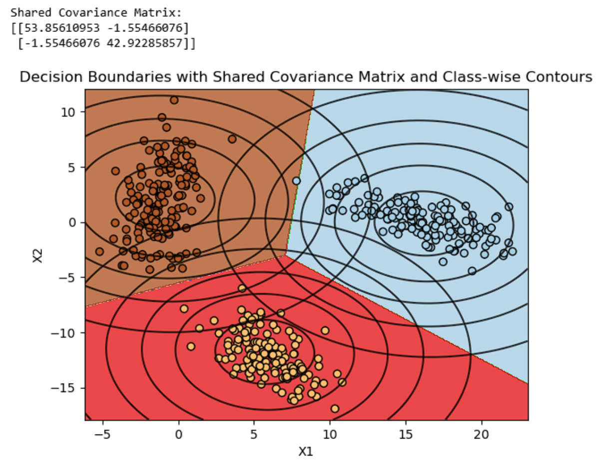

Fig. 5. Full Covariance Matrix and Decision Boundary for Linearly Separable Data and Class-wise Gaussian Contours

- *Case 3: Arbitrary Covariance Matrix and Decision Boundary*

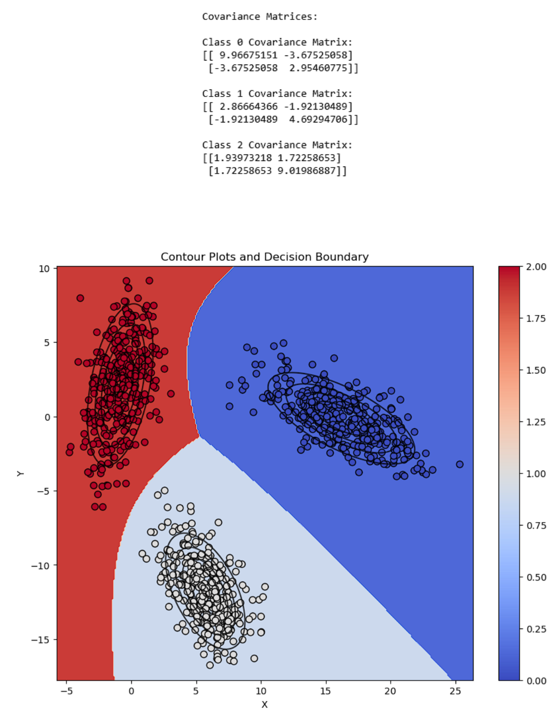

Fig. 6. Arbitrary Covariance Matrix and Decision Boundary for Linearly Separable Data and Class-wise Gaussian Contours

2) *Performance Evaluation Metrics*: Evaluation metrics, including accuracy, precision, recall, and F-score, are computed to assess the classifier's performance. The confusion matrix provides detailed insights into misclassifications, which are minimal in this scenario.

```
Confusion Matrix:
[[148   0   0]
 [  0 141   0]
 [  0   0 161]]

Accuracy: 1.00

Precision for each class:
Class 0: 1.00
Class 1: 1.00
Class 2: 1.00

Recall for each class:
Class 0: 1.00
Class 1: 1.00
Class 2: 1.00

F-score for each class:
Class 0: 1.00
Class 1: 1.00
Class 2: 1.00

Mean Precision: 1.00
Mean Recall: 1.00
Mean F-score: 1.00
```

Fig. 7. Performance Metrics for Linearly Separable Data

B. Non-linear Dataset

Non-linear datasets involve classes with overlapping distributions, often requiring more flexible covariance structures to model feature correlations.

1) *Covariance Analysis and Decision Boundary*: Full covariance matrices are calculated for each class to capture interdependencies between features. These matrices directly influence the shape of the decision boundaries, which are visualized using contour plots. The resulting boundaries, often quadratic or higher-order, illustrate the classifier's ability to handle complex and overlapping class distributions, offering a detailed view of the regions predicted for each class.

   - *Case 1: Isotropic Covariance Matrix and Decision boundary*

```
Shared Isotropic Covariance Matrix:
[[6.22018907 0.        ]
 [0.         6.22018907]]
```

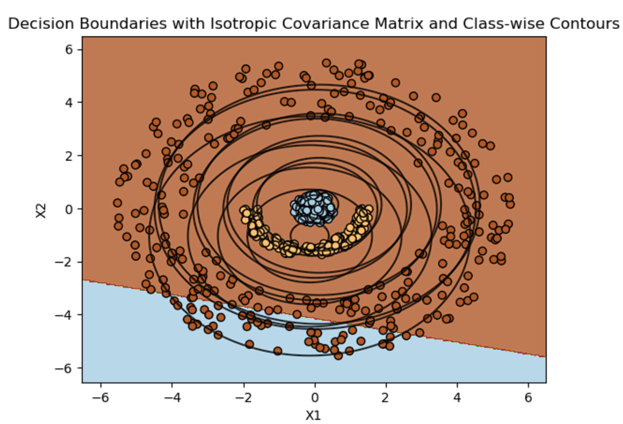

Fig. 8. Isotropic Covariance Matrix and Decision Boundary for Non-Linearly Separable Data and Class-wise Gaussian Contours

   - *Case 2: Full Covariance Matrix with identical Covariance Pairs but Different Variances and Decision Boundary*

```
Shared Isotropic Covariance Matrix:
[[6.22018907 0.        ]
 [0.         6.22018907]]
```

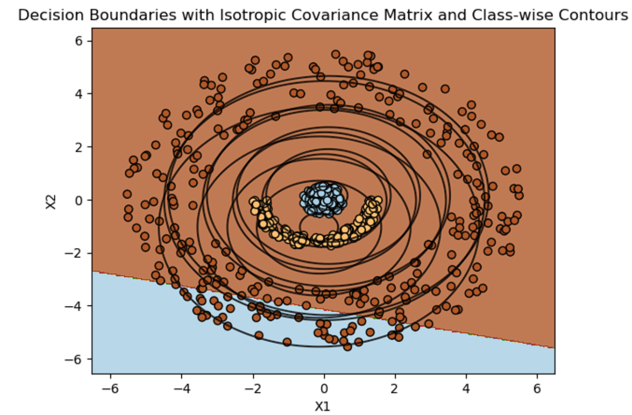

Fig. 9. Full Covariance Matrix and Decision Boundary for Non-Linearly Separable Data and Class-wise Gaussian Contours

   - *Case 3: Arbitrary Covariance Matrix and Decision Boundary*

```
Covariance Matrices:

Class 0 Covariance Matrix:
[[ 0.09512208 -0.01123553]
 [-0.01123553  0.09983244]]

Class 1 Covariance Matrix:
[[1.21589522 0.02775699]
 [0.02775699 0.26574453]]

Class 2 Covariance Matrix:
[[10.02115828  0.24632415]
 [ 0.24632415 11.20806045]]
```

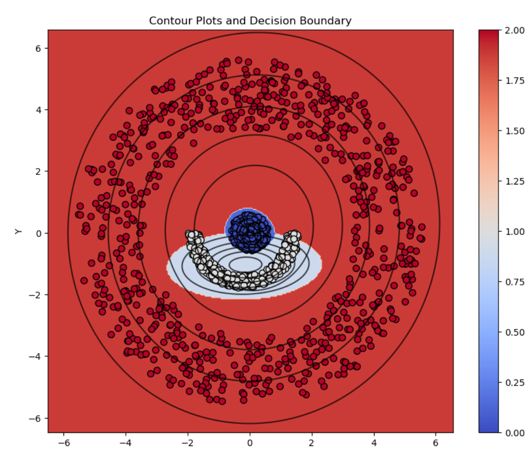

Fig. 10. Arbitrary Covariance Matrix and Decision Boundary for Non-Linearly Separable Data and Class-wise Gaussian Contours

2) *Performance Evaluation Metrics*: The evaluation is conducted using the same metrics as before. Slightly lower accuracy compared to the linear dataset may be observed due to the increased complexity of class separation. However, F-score provides a balanced view of precision and recall performance.

```
Confusion Matrix:
[[ 80   0   0]
 [  0 140  17]
 [  0   0 303]]

Accuracy: 0.97

Precision for each class:
Class 0: 1.00
Class 1: 1.00
Class 2: 0.95

Recall for each class:
Class 0: 1.00
Class 1: 0.89
Class 2: 1.00

F-Score for each class:
Class 0: 1.00
Class 1: 0.94
Class 2: 0.97

Mean Precision: 0.98
Mean Recall: 0.96
Mean F-Score: 0.97
```

Fig. 11. Performance Metrics for Non-Linearly Separable Data

C. Real-World Dataset

For the real-world dataset, the classes are modeled without assumptions about linearity or specific distributions, ensuring generalizability of the classifier.

1) *Covariance Analysis and Decision Boundary*: Each class is represented by a unique covariance matrix, capturing real-world variability and feature correlations. Depending on the data, both linear and non-linear decision boundaries are observed. These boundaries, visualized over real-world data, provide insights into how the classifier segregates natural distributions, including overlapping regions and well-separated classes.

- *Case 1: Isotropic Covariance Matrix and Decision boundary*

```
Shared Isotropic Covariance Matrix:
[[241979.15559335      0.        ]
 [     0.         241979.15559335]]
```

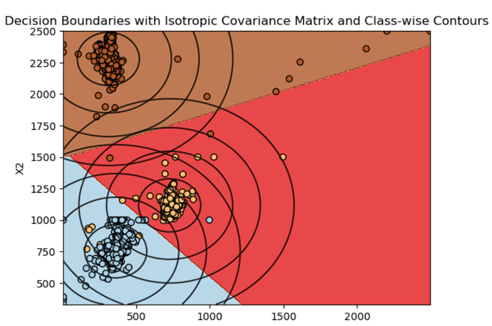

Fig. 12. Isotropic Covariance Matrix and Decision Boundary for Real World Data and Class-wise Gaussian Contours

- *Case 2: Full Covariance Matrix with identical Covariance Pairs but Different Variances and Decision Boundary*

```
Shared Covariance Matrix:
[[ 40228.73407571 -44146.97553748]
 [-44146.97553748 443827.80286929]]
```

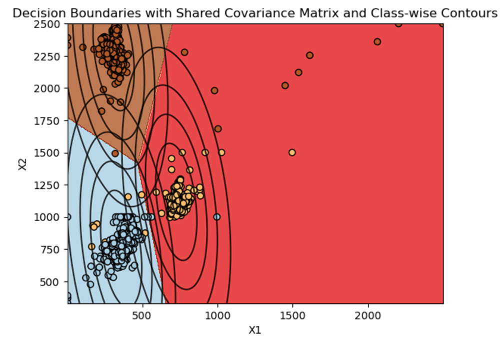

Fig. 13. Full Covariance Matrix and Decision Boundary for Real World Data and Class-wise Gaussian Contours

- *Case 3: Arbitrary Covariance Matrix and Decision Boundary*

```
Covariance Matrices:

Class 0 Covariance Matrix:
[[3855.90517846 3272.22106917]
 [3272.22106917 7466.84482816]]

Class 1 Covariance Matrix:
[[6923.77112787 3314.5250532 ]
 [3314.5250532  6053.31729729]]

Class 2 Covariance Matrix:
[[10449.60724403  -545.70274935]
 [ -545.70274935 12773.31559286]]
```

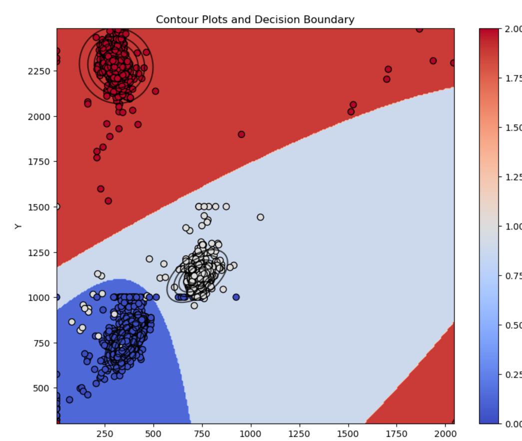

Fig. 14. Arbitrary Covariance Matrix and Decision Boundary for Real World Data and Class-wise Gaussian Contours

2) *Performance Evaluation Metrics*: A comprehensive analysis using the same metrics is conducted. The confusion matrix highlights areas of improvement and potential biases in the classifier's predictions, crucial for real-world applications.

```
Confusion Matrix:
[[769   6   0]
 [ 16 620   0]
 [  0   1 700]]

Accuracy: 0.99

Precision for each class:
Class 0: 0.98
Class 1: 0.99
Class 2: 1.00

Recall for each class:
Class 0: 0.99
Class 1: 0.97
Class 2: 1.00

F-Score for each class:
Class 0: 1.00
Class 1: 1.00
Class 2: 1.00

Mean Precision: 0.99
Mean Recall: 0.99
Mean F-Score: 1.00
```

Fig. 15. Performance Metrics for Real-World Data

V. Key Takeaways on Covariance Effects on Decision Boundaries

1. Isotropic Covariance Matrix ($\Sigma = \sigma^2 I$)

- **Covariance Matrix:** The isotropic covariance matrix assumes that all features have the same variance (σ^2), and there are no covariances (all off-diagonal elements are zero). This configuration represents a uniform spread of data in all directions around the mean vector. Mathematically:

$$\Sigma = \begin{bmatrix} \sigma^2 & 0 & \dots & 0 \\ 0 & \sigma^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma^2 \end{bmatrix}$$

- **Contours:** The contours for the Gaussian distribution are perfect circles (or spheres in higher dimensions), indicating identical spread in all directions. These contours do not adjust to data elongations or skewness.
- **Decision Boundary:**
  - The decision boundaries are linear and symmetric if class priors are equal.
  - When priors differ, the linear decision boundary shifts towards the class with higher variance.
  - Simplicity makes it ideal for linearly separable data but unsuitable for skewed or elongated distributions.
- **Key Observations:** The isotropic covariance assumption is computationally efficient but lacks flexibility. While it is effective for well-separated, uniformly distributed data, it struggles to model real-world datasets with varying feature correlations and variances.

2. Full Covariance Matrix (Unequal Variances, Equal Non-zero Covariances)

- **Covariance Matrix:** A full covariance matrix assumes that each feature has its own variance, and there exist equal and non-zero covariances (ρ) between feature pairs. This representation models interdependencies between features effectively:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho & \dots & \rho \\ \rho & \sigma_2^2 & \dots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \dots & \sigma_d^2 \end{bmatrix}$$

- **Contours:** The contours of the Gaussian distribution are tilted ellipses, reflecting the correlation between features.
- **Decision Boundary:**
  - The boundaries are quadratic or skewed, accurately reflecting feature correlations and class distributions.
  - More flexible than isotropic covariance but simpler compared to arbitrary covariance structures.
- **Key Observations:** Full covariance effectively captures dependencies between features, resulting in realistic and adaptable decision boundaries suitable for correlated data.

3. Arbitrary Covariance Matrix (General Case)

- **Covariance Matrix:** The arbitrary covariance matrix allows for full flexibility, with unique variances for each feature and non-zero, distinct covariances. This general representation provides the highest adaptability to data structures.
- **Contours:** Contours are ellipses (or ellipsoids in higher dimensions) with arbitrary orientation and shape, capturing the full complexity of feature distributions.
- **Decision Boundary:**
  - Boundaries are fully flexible and typically quadratic, enabling them to adapt to highly complex data distributions.
  - Suitable for challenging datasets with intricate feature relationships and overlapping class regions.
- **Key Observations:** While the arbitrary covariance matrix provides the most flexibility, it is computationally expensive and can lead to overfitting in scenarios with limited data.