Mathematics for Data Science - DSL201

ASSIGNMENT 1

Siddhi Singh, B24DS508

Question 1

Introduction

The Central Limit Theorem (CLT) is a key concept in statistics, stating that the distribution of sample means approaches a normal distribution as the sample size increases, regardless of the population's original shape.

In this report, the CLT is demonstrated using two non-normal populations: a Uniform distribution and an Exponential distribution. For each case, repeated random samples of varying sizes are drawn, their means are computed, and the resulting sampling distributions are compared with the original population. The results show that as sample size increases, the sampling distributions become increasingly symmetric and bell-shaped, illustrating the CLT in practice.

Methodology

To demonstrate the effect of the Central Limit Theorem (CLT) on non-Gaussian distributions, we follow a structured process with separate functions for each step to ensure clarity. First, the original distribution is generated using NumPy's random library, which provides built-in functions for uniform and exponential distributions. These are plotted as histograms of values versus density using Seaborn. Density is used instead of raw counts, as it highlights the shape of the distribution more effectively by normalizing the scale.

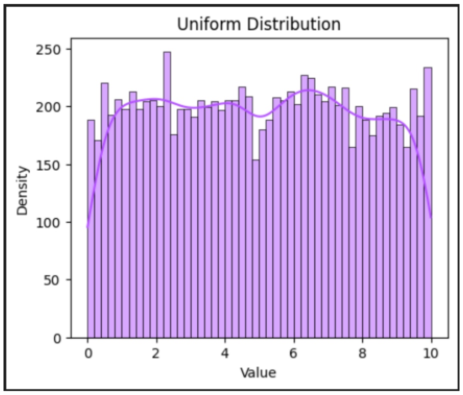
Next, repeated samples of different sizes are drawn, and their means are calculated. This step represents the core idea of the CLT: regardless of the population's shape, the distribution of sample means tends toward normality as the sample size increases.

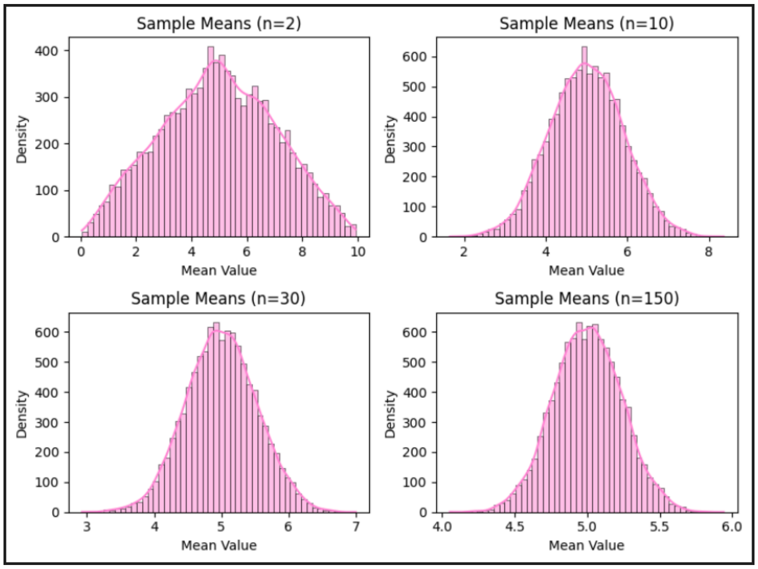
Finally, the sampling distributions are plotted as histograms of mean values versus density for various sample sizes (e.g., n = 2, 10, 30, 150). These visualizations show how the initially skewed or uniform distributions progressively take on the bell-shaped form of a normal distribution as more sample means are considered. All the functions are then combined to produce a complete demonstration of the CLT.

Results

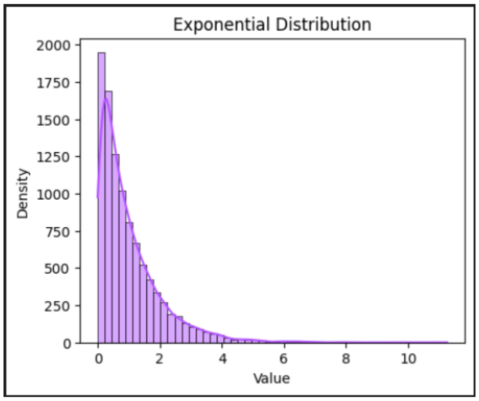
Histograms of the Uniform distribution are shown below.

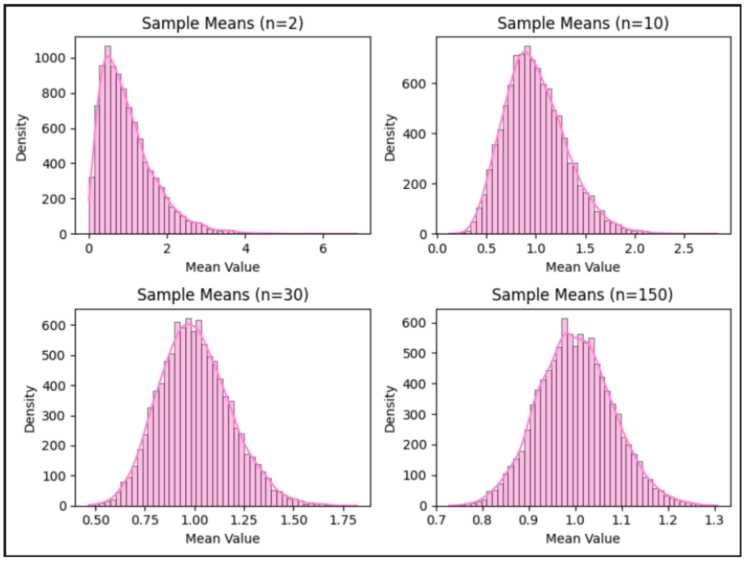
The graph on the right represents the original distribution, while the graphs on the left show the sampling distributions obtained after taking means of samples.





Histograms of the Exponential distribution are shown below.





Discussion

We observe that when the sample size is very small, such as when only two observations are averaged, the resulting sampling distribution does not resemble the bell-shaped curve of a Gaussian distribution. In this case, the histograms remain closer in shape to the original distribution, whether uniform or exponential, and lack the symmetry associated with a normal curve. However, as the sample size increases and more sample means are calculated, a clear transformation occurs. The distributions of the means gradually take on a bell-shaped form and become increasingly symmetric, demonstrating the behavior predicted by the Central Limit Theorem.

This shift highlights the fundamental idea that, regardless of the original distribution's form, the process of repeatedly averaging random samples produces a distribution of means that converges toward normality. Another important observation is that the spread of the sampling distribution decreases as the sample size increases. In other words, the variability of the sample means becomes smaller, and the means are more tightly clustered around the true population mean. This reduction in range reflects the fact that larger samples provide more stable and reliable estimates of the population mean, further emphasizing the practical significance of the Central Limit Theorem.

Conclusion

Through this experiment, we have demonstrated the Central Limit Theorem (CLT) using two non-normal populations: the Uniform and Exponential distributions. In both cases, the results confirm the theoretical prediction that the distribution of sample means approaches a normal distribution as the sample size increases. For small sample sizes, the sampling distributions closely resemble the shape of the original populations, lacking the symmetry of a Gaussian curve. However, as the number of observations per sample grows, the distributions of the means become increasingly bell-shaped and symmetric.

Additionally, it was observed that the variability of the sample means decreases with larger sample sizes, leading to a narrower range of values centered around the true population mean. This illustrates the practical importance of the CLT: it ensures that, regardless of the underlying population distribution, the sampling distribution of the mean will converge to normality, making statistical inference possible even for non-Gaussian data.

Mathematics for Data Science - DSL201

ASSIGNMENT 1

Siddhi Singh, B24DS508

Question 2

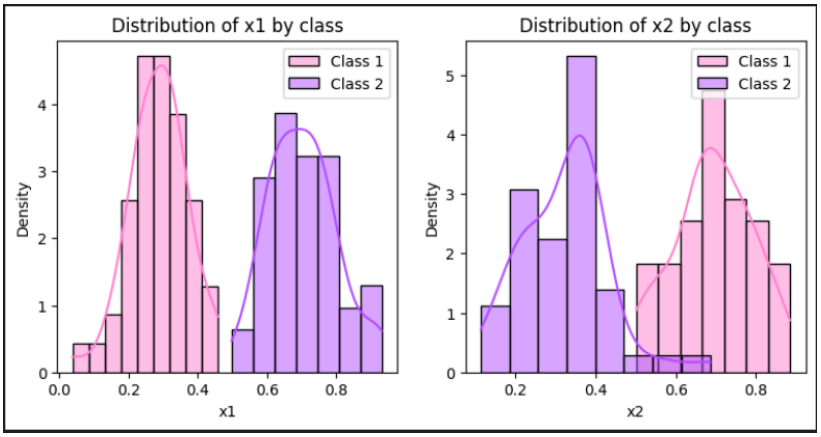
Introduction

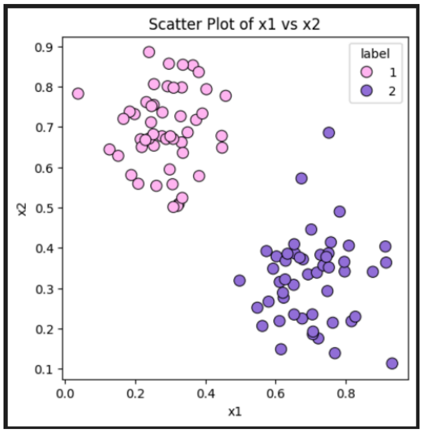
In this task, we are provided with two datasets, each divided into training and test sets. Every dataset consists of two features, x1 and x2, along with a label column indicating class 1 or 2. The objective is to apply Bayes' Theorem in combination with Maximum Likelihood Estimation (MLE) to perform classification. The approach begins by estimating the class-specific parameters: the means (μ1, μ2), covariance matrices (Σ1, Σ2), and prior probabilities (c1, c2) from the training data. Using these parameters, Bayes' Theorem is then applied to compute the posterior probabilities for the test points. Each test sample is classified into the class with the higher posterior probability. To evaluate the performance of the classifier, the classified test data is visualized, and the overall accuracy score is computed.

Data

Both datasets consist of 100 entries each, with no missing values. Each record contains two features, x1 and x2, along with a class label taking the value 1 or 2. The feature values for x1 and x2 lie within the range [0, 1]. To apply Maximum Likelihood Estimation (MLE), it is first necessary to examine the underlying data distribution. For this purpose, class-wise histograms and scatter plots were generated. The resulting visualizations for Dataset 1 and Dataset 2 are presented below.

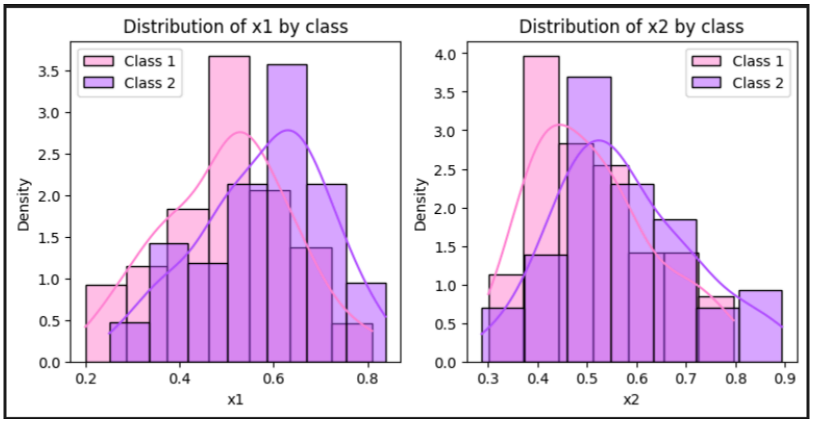
For dataset 1:

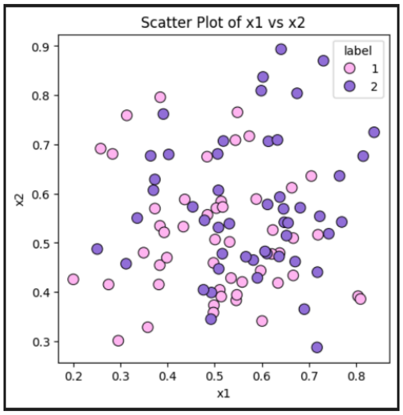




The histograms for both classes exhibit a bell-shaped pattern, indicating that the data follows a Gaussian distribution. Similarly, the scatter plots show elliptical clusters for each class, further supporting this assumption. Since the overlap between the two classes is minimal, the Bayes classifier is expected to achieve high accuracy on this dataset.

For dataset 2:





For this dataset, the histograms reveal skewed distributions, and the scatter plot shows significant overlap between the two classes. This indicates that applying Bayes' classifier may lead to a higher rate of misclassification.

Methodology

To apply Bayes' Theorem, we first estimate the class means (μ_1, μ_2) assuming Gaussian distributions. The Maximum Likelihood Estimate (MLE) of the mean is:
$\mu = (1/n)\ \Sigma$ (from i=1 to n) x_i

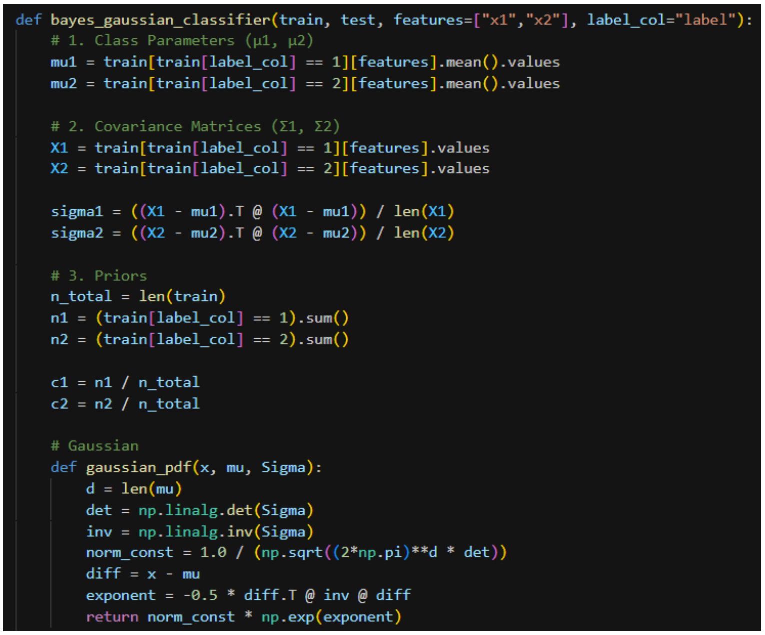
Using this formula, the means for each class are calculated. Next, the covariance matrices are computed using:
$\Sigma = (1/n)\ \Sigma$ (from i=1 to n) $(x_i - \mu)(x_i - \mu)^T$

The prior probabilities are then determined. Since the number of samples in class 1 and class 2 is equal, the priors are:
P(label=1) = P(label=2) = 0.5

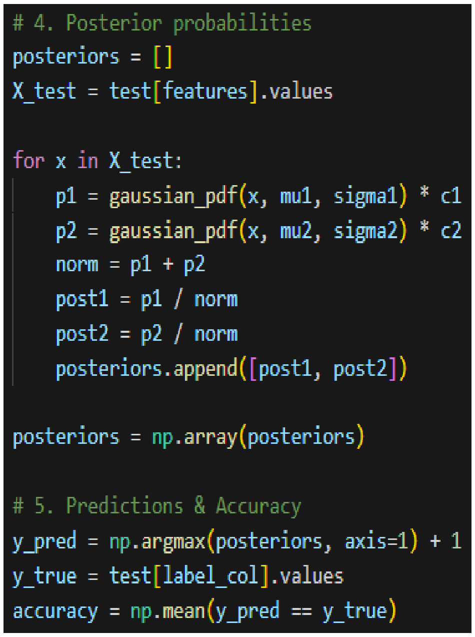
With the means, covariances, and priors obtained, we calculate the posterior probabilities for each test point (x_1, x_2) using Bayes' theorem:
$P(C\square \mid x) = [\ p(x \mid C\square) \cdot P(C\square)\ ]\ /\ \Sigma$ (over j=1 to 2) $[\ p(x \mid C\square) \cdot P(C\square)\ ]$

Each test sample is then assigned to the class with the higher posterior probability:
$\hat{y} = \text{argmax}$ (over $k \in \{1, 2\}$) $P(C\square \mid x)$

A new column "predicted" is created to store the assigned labels. The classification accuracy is measured by comparing the predicted labels with the true labels. Finally, scatter plots of both the true and predicted labels (x_1 vs. x_2) are generated to visually assess the performance of the Bayesian classifier.



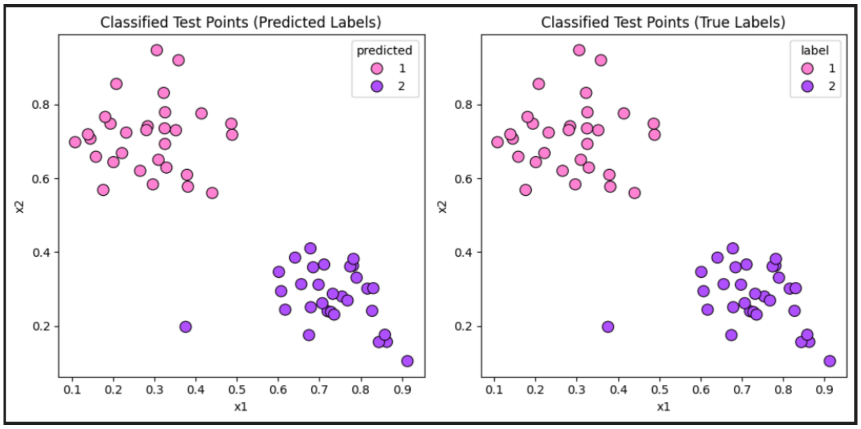
```python
def bayes_gaussian_classifier(train, test, features=["x1","x2"], label_col="label"):
    # 1. Class Parameters (μ1, μ2)
    mu1 = train[train[label_col] == 1][features].mean().values
    mu2 = train[train[label_col] == 2][features].mean().values

    # 2. Covariance Matrices (Σ1, Σ2)
    X1 = train[train[label_col] == 1][features].values
    X2 = train[train[label_col] == 2][features].values

    sigma1 = ((X1 - mu1).T @ (X1 - mu1)) / len(X1)
    sigma2 = ((X2 - mu2).T @ (X2 - mu2)) / len(X2)

    # 3. Priors
    n_total = len(train)
    n1 = (train[label_col] == 1).sum()
    n2 = (train[label_col] == 2).sum()

    c1 = n1 / n_total
    c2 = n2 / n_total

    # Gaussian
    def gaussian_pdf(x, mu, Sigma):
        d = len(mu)
        det = np.linalg.det(Sigma)
        inv = np.linalg.inv(Sigma)
        norm_const = 1.0 / (np.sqrt((2*np.pi)**d * det))
        diff = x - mu
        exponent = -0.5 * diff.T @ inv @ diff
        return norm_const * np.exp(exponent)
```

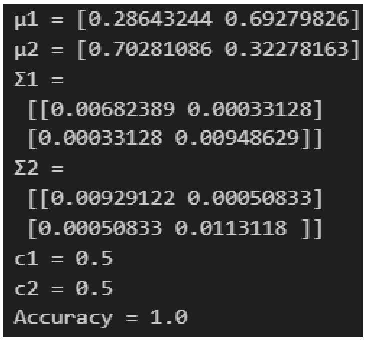


```python
# 4. Posterior probabilities
posteriors = []
X_test = test[features].values

for x in X_test:
    p1 = gaussian_pdf(x, mu1, sigma1) * c1
    p2 = gaussian_pdf(x, mu2, sigma2) * c2
    norm = p1 + p2
    post1 = p1 / norm
    post2 = p2 / norm
    posteriors.append([post1, post2])

posteriors = np.array(posteriors)

# 5. Predictions & Accuracy
y_pred = np.argmax(posteriors, axis=1) + 1
y_true = test[label_col].values
accuracy = np.mean(y_pred == y_true)
```

Results

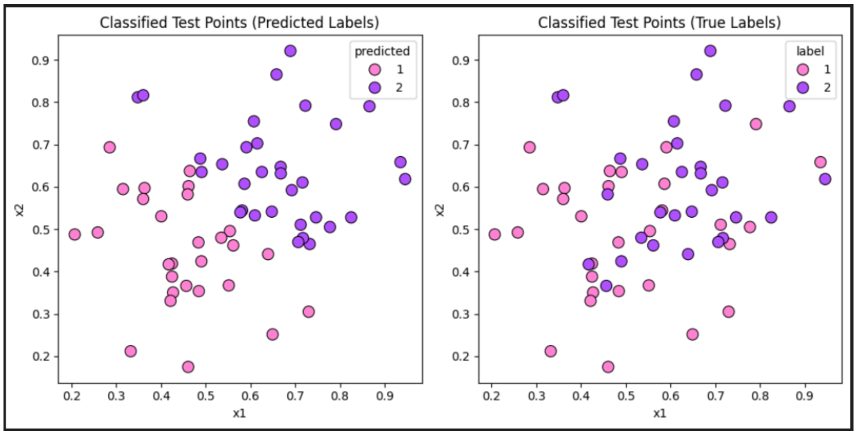
For Dataset 1, the classifier achieved 100% accuracy. The results are shown below.

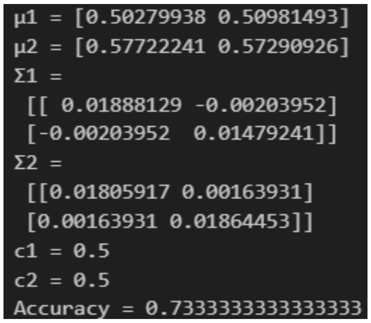




```
μ1 = [0.28643244 0.69279826]
μ2 = [0.70281086 0.32278163]
Σ1 =
 [[0.00682389 0.00033128]
 [0.00033128 0.00948629]]
Σ2 =
 [[0.00929122 0.00050833]
 [0.00050833 0.0113118 ]]
c1 = 0.5
c2 = 0.5
Accuracy = 1.0
```

For Dataset 2, the classifier achieved 73.3% accuracy. The results are shown below.





```
μ1 = [0.50279938 0.50981493]
μ2 = [0.57722241 0.57290926]
Σ1 =
 [[ 0.01888129 -0.00203952]
 [-0.00203952  0.01479241]]
Σ2 =
 [[0.01805917 0.00163931]
 [0.00163931 0.01864453]]
c1 = 0.5
c2 = 0.5
Accuracy = 0.7333333333333333
```

Discussion

In Dataset 1, the feature distributions are approximately Gaussian, with minimal overlap between the classes and mean values well separated from each other. This is reflected in the results, where the classifier achieves 100% accuracy, and the scatter plot shows perfect separation. The covariance values are very low, which validates the assumption of feature independence used in Bayes' theorem.

In Dataset 2, however, the distributions are skewed and exhibit noticeable overlap, with the class means situated close to each other. Consequently, the classifier's performance is affected, resulting in 73.3% accuracy. While the covariances are still low, they are slightly higher than in Dataset 1. This suggests that the assumption of feature independence in the Naive Bayes approach may be contributing to the reduced accuracy

Conclusion

In this project, we applied Bayes' Theorem combined with Maximum Likelihood Estimation (MLE) to classify points in two datasets. Dataset 1 exhibited near-Gaussian distributions with well-separated classes and minimal overlap, resulting in perfect classification (100% accuracy). Dataset 2 showed skewed distributions and overlapping classes, which led to a reduced accuracy of 73.3%, highlighting the impact of feature dependence and overlap on the Naive Bayes assumption. Overall, the study demonstrates that Bayes' classifier performs best when class distributions are approximately Gaussian, classes are well separated, and feature independence holds. Visualization of the classified points provided clear insight into model performance and the relationship between class distributions and accuracy.