Date of Submission June 2024

Digital Object Identifier Not Applicable

A Study on the Implementation of Principal Component Analysis and Linear Discriminant Analysis

NIMESH GOPAL PRADHAN[1], RAMAN BHATTARAI[1]
[1]Department of Electronics and Computer Engineering, Thapathali Campus, Tribhuvan University, Kathmandu, Nepal

ABSTRACT Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are powerful techniques for dimensionality reduction and data exploration in multi-dimensional datasets. This paper presents a comprehensive examination of both PCA and LDA, detailing their fundamental principles and practical applications. The implementation process of PCA is outlined, starting with the standardization of the data, followed by the calculation of covariance matrices, eigenvalues, and eigenvectors. The selection of principal components and subsequent data transformation are then discussed. PCA offers several benefits, including noise reduction, improved visualization, and enhanced analytical capabilities. Additionally, the paper delves into the principles and application of LDA, which differs from PCA by incorporating class labels in the dimensionality reduction process. LDA aims to maximize the separation between classes while minimizing the variance within each class. The implementation of LDA involves similar steps to PCA, with a focus on finding linear discriminants that best separate the classes. Furthermore, this research investigates the differences between PCA and LDA in terms of their effectiveness in dimensionality reduction and their impact on classification performance. By comparing the results of applying both techniques to the same dataset, this study aims to provide insights into when to use PCA or LDA based on the characteristics of the data and the objectives of the analysis.
By explaining the step-by-step execution of both PCA and LDA, this paper provides valuable insights for researchers and practitioners in various fields such as image processing, data mining, and pattern recognition. The ability of PCA and LDA to uncover underlying patterns and relationships in complex datasets makes them essential tools in data analysis and interpretation.

INDEX TERMS Covariance, eigenvalue, eigenvector, linear discriminant analysis, multi-dimensional datasets, principal component analysis, scatter matrix, variance.

I. INTRODUCTION

A. MULTI-DIMENSIONAL DATASETS

Multi-Dimensional datasets are datasets comprised of observations characterized by multiple attributes or features, commonly found in various fields such as statistics, machine learning, and data analysis.

B. VARIANCE

Variance is a measure of the dispersion or spread of data points around their mean value, which is essential for determining the importance of each principal component in PCA and guiding feature selection and interpretation.

C. COVARIANCE

Covariance is a measure of the degree to which two variables change together, indicating their relationship and dependency, utilized in PCA and LDA for capturing the relationships between different features or classes and improving classification accuracy.

D. EIGENVALUE

Eigenvalue is a scalar value that represents the scaling factor by which an eigenvector is stretched or compressed when subjected to a linear transformation, it is crucial for determining the significance of principal components or discriminant directions in PCA and LDA.

E. EIGENVECTOR

Eigenvector is a vector that remains unchanged in direction when subjected to a linear transformation, which defines the principal axes along which data varies the most in PCA or the discriminant directions in LDA.

F. WITH-IN CLASS SCATTER MATRIX

It is a matrix that quantifies the dispersion of data points within individual classes. This matrix helps in maximizing the compactness of data points within each class.

G. BETWEEN CLASS SCATTER MATRIX

It is a matrix that quantifies the dispersion between different classes. This matrix helps in maximizing the separation between class means in the feature space.

H. PRINCIPAL COMPONENT ANALYSIS (PCA)

Principle Component Analysis (PCA), introduced by Pearson in 1901, is a classical and widely adopted method for dimensionality reduction, effectively simplifying complex datasets while retaining essential information. PCA achieves this by identifying orthogonal directions, known as principal components, that capture the maximum variance within the data. The first principal component represents the direction of maximum variance, and each subsequent component captures the remaining variance orthogonal to the previous ones.

This method is crucial for addressing the challenges posed by high-dimensional datasets. While having high-dimensional data can help in machine learning models discriminate the instances into classes pretty easily, it also poses some problems such as increased computational requirements and the presence of noise. By reducing dimensionality, PCA enhances computational efficiency and data interpretability, making it easier to visualize and analyze data in lower-dimensional spaces like 1D, 2D or 3D. PCA can also remove noise and repetitive information present in high-dimensional data hence putting more focus on the most informative components of the data. Understanding PCA provides a foundational knowledge of dimensionality reduction techniques, necessary to tackle complex data analysis challenges and optimize data representation.

PCA revolves around identifying the main aspects of data known as principal components. These components are essentially perpendicular vectors that grab the most variance in the data. Their perpendicularity means they're independent and don't relate to each other. This independence lets PCA catch various sides of the data's differences, offering a useful view of the data in a smaller space. This condensed space keeps the most important information while ditching less crucial information, effectively reducing the dimensionality of the dataset.

This paper offers an in-depth examination of PCA on two different datasets, detailing its implementation through standardizing data, calculating covariance matrices, extracting eigenvalues and eigenvectors, selecting principal components, and transforming data. Firstly, PCA has been conducted on a randomly created dataset and then on the Palmer Penguins dataset. Through this exploration, the paper highlights the benefits and practical applications of PCA, serving as a valuable resource for those seeking to leverage this influential tool in various domains such as image processing, data mining, and pattern recognition.

I. LINEAR DISCRIMINANT ANALYSIS (LDA)

Linear Discriminant Analysis (LDA) is another powerful technique for dimensionality reduction and classification, distinct from PCA in that it incorporates class labels during the dimensionality reduction process. Introduced by R. A. Fisher in 1936, LDA aims to find the linear combinations of features that best separate different classes in the data.

Unlike PCA, which focuses solely on maximizing the variance in the dataset, LDA considers both the variance within each class and the variance between different classes. By maximizing the separation between classes while minimizing the variance within each class, LDA identifies the directions in the feature space that are most effective for discriminating between classes.

This paper will also explore the principles and application of LDA, detailing its implementation process and practical implications. By comparing the results of PCA and LDA on the Palmer Penguin dataset, this study aims to provide insights into the effectiveness of different dimensionality reduction techniques for various data analysis tasks.

II. RELATED WORKS

Principal component analysis (PCA) is linear dimensionality reduction technique used to study multivariate data. It is still used in fields as dimensionality reduction, features extraction and research due to its versatility. The book of Jolliffe "Principal Component Analysis" provides insight on PCA, with its implementation, application, mathematical principles and current research [1]. PCA is the foundation for modern data analysis. Shlens paper "A Tutorial on Principal Component Analysis" eliminates the obscure concept about PCA and provides ideas how and why PCA works with the help of basic, fundamental ideas and mathematics equations and principles of PCA [2]. PCA is usually needed for analyzing data in distinct areas. This paper "Principal Component Analysis: A Natural Approach to Data Exploration" explains basic principles of PCA, data standardization, possible visualization of PCA. It also show dimensionality reduction on various real-world datasets using PCA and other dimensionality reduction techniques [3]. PCA improves understandability as well as reduce information loss of datasets. It does this by calculating eigenvalue/eigenvectors to obtain principal components and creating new uncorrelated variables defined by the available datasets. This paper "Principal component analysis: a review and recent developments" illustrates PCA as an adaptive data analysis technique, explain some of its variants together with their applications [4]. Linear discriminant analysis (LDA) is designed to find an optimal transformation to extract discriminant features that characterize two or more classes of objects. Finally, through experiments, this article "Linear Discriminant Analysis" we show how LDA can be used in face recognition [5]. The aim of this paper "Linear discriminant analysis: A detailed tutorial" is to build a solid intuition for what is LDA, and how LDA works, thus enabling readers of all levels be able to get a better understanding of the LDA and to know how to apply this technique in different applications. Moreover, the two methods of computing the LDA space, i.e. class-dependent and class-independent methods, were explained in details [6]. Linear Discriminant Analysis easily handles the case where the within-class frequencies are unequal and their performances has been examined on randomly generated test data. This paper "Linear Discriminant Analysis—A Brief Tutorial" explains the prime difference between LDA and PCA that is PCA does more of feature classification and LDA does data classification [7].

III. METHODOLOGY

A. DATASET INFORMATION

The datasets used in this paper are a random dataset and Palmer Penguins dataset. The random dataset was obtained by randomly sampling 20 points from a Gaussian distribution. This dataset has 20 instances of data with 2 attributes each. So the dimension of the data matrix is 20×2. In the Palmer Penguins dataset there are 344 instances with 6 attributes and 1 target attribute. The six attributes are island, bill length, bill depth, flipper length, body mass and sex while the target attribute, species contains three penguin varieties: Adelie, Chinstrap and Gentoo. After filtering the rows with missing values the dataset now comprised a total of 333 instances The 333 instances of data is distributed as follows 146 instances of Adelie penguins, 119 instances of Chinstrap penguins, and 68 instances of Gentoo penguins. Some of the features which were not numeric like island and sex were omitted from the PCA analysis. Thus, the data matrix for the Palmer Penguins dataset is of dimension 333×4.

B. WORKING MECHANISM

1) Preprocessing of Dataset

Consider, a dataset, in our case it would be the Palmer Penguins dataset. The first step is to pre-process the dataset in order to remove the missing values and encode the categorical feature into a numerical one. The dataset is then represented as a matrix Z of size $n \times m$. The matrix can be expressed as

$$Z_{m\times n} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1} & z_{m2} & \cdots & z_{mn} \end{bmatrix} \tag{1}$$

The second step is to standardize the data. This involves calculating the mean and standard deviation for each dimension. Each element in a given dimension is then centered by subtracting the mean, and subsequently scaled by dividing by the standard deviation. This ensures that the data in each dimension has a mean of zero and a standard deviation of one. This is done to ensure that each feature contributes equally to the analysis, regardless of their original scales. The standardized data X is calculated using the formula:

$$X = \frac{Z-\mu}{\sigma} \tag{2}$$

where:

- Z is the original data.
- X is the standarized data.
- μ is the mean of each feature.
- σ is the standard deviation of each feature.

The size of the matrix is unchanged by above operation.

2) Working Mechanism of PCA

In PCA, after preprocessing the dataset the covariance matrix of X is calculated. This calculation is done using the formula:

$$S_X = \frac{1}{n-1} X^T \cdot X \tag{3}$$

where:

- S_X is the covariance matrix.
- X is the standardized data matrix.
- X^T is the transpose matrix of X
- n is the number of observations.

This covariance matrix is used in determining the eigenvalues and eigenvectors. The size of S_X is now $m \times m$ i.e a square matrix. Now, the eigenvalues can be calculated by solving the characteristics equation below:

$$|S_X - \lambda \cdot I| = 0 \tag{4}$$

where:

- λ is a scalar (eigenvalue)
- I is an identity matrix having same size as S_X

If the eigenvalues are found to be $\lambda_1, \lambda_2, \ldots, \lambda_n$, the respective eigenvectors is obtained from the following equation:

$$(S_X - \lambda_i \cdot I)\mathbf{v}_i = 0 \tag{5}$$

where $\mathbf{v}_i$ is the eigenvector corresponding to eigenvalue λ_i.

The eigenvectors of the covariance matrix serves as the principal components of the datasets indicating the direction in the original feature space with the highest variability. Corresponding eigenvalues signify the amount of variance explained by each principal component. Arranging the eigenvectors in descending order of their eigenvalues helps establish their significance. Once the eigenvalues and eigenvectors are obtained, a subset of eigenvector is selected to represent the dataset. This selection enables dimensionality reduction through the projection of original data onto chosen eigenvectors, thereby preserving most of the information while discarding less significant components. For datasets with m dimensions, there are m eigenvalues and eigenvectors. By selecting only the first r most significant eigenvectors, the dataset's dimensions reduces to r. The proportion of variance explained by these r eigenvalues can be computed using the formula:

$$\text{Proportion of Variance} = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{r} \lambda_i} \tag{6}$$

This is the measure of how much these components account for the total variance. If the dimensions are highly correlated , there will be a small number of eigenvectors with large eigenvalues and r will be much smaller than m. If the dimensions are not correlated, r will be as large as m and PCA does not help

Now, the final matrix Y is found using the equation:

$$Y = P \cdot X \tag{7}$$

Where, P is the row feature vector which is a matrix with the eigenvectors in the columns transposed so that the eigenvectors are now in the rows, with the most significant eigenvector at the top. and Y is simply the original data in terms of the vectors we choose. The covariance matrix of Y is calculated by:

$$S_Y = \frac{1}{n-1} Y \cdot Y^T \tag{8}$$

The covariant matrix provides insights into the relationships between different dimensions and is always a square matrix. The diagonal elements of this matrix represent the variances of the principle components, while the off-diagonal elements represent the covariances between different variables. The goal is to minimize the off-diagonal values, ideally bringing them to zero, and to maximize the diagonal values as much as possible. Thus, the covariance matrix is crucial in assessing whether the selected principal components are optimal.

The exact number of principal components required is uncertain, but the objective is to minimize the number retained while maximizing the accuracy of data representation. One method to achieve this is by evaluating the proportion of the variance each principal component explains. This is expressed by calculating the percentage of the total variance accounted for by the i^{th} eigenvector as follows:

$$r_i = 100 \times \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \tag{9}$$

One heuristic method is to discard components where r_i falls below a threshold like 1%.

3) Working Mechanism of LDA

In LDA, after preprocessing the dataset, we calculate the mean vectors μ_c of all the present classes which provides a representative measure of the central tendency of each class by using the formula:

$$\mu_c = \frac{1}{N_c} \sum_{i=1}^{N_c} x_i^c \tag{10}$$

Where:

- μ_c represents the mean vector of class c.
- N_c is the number of samples in class c.
- x_i^c is the feature vector of the i^{th} sample in class c.

Following this, the within-class scatter matrix S_W is calculated, capturing the spread of data within each class. It quantifies the variations within individual classes, necessary for distinguishing between different classes. It is computed by summing up the covariance matrices of each class, weighted by their respective class sizes. Mathematically, it is represented as:

$$S_W = \sum_{c=1}^{C} \sum_{i=1}^{N_c} (x_i^c - \mu_c)(x_i^c - \mu_c)^T \tag{11}$$

Where:

- S_W is the within-class scatter matrix.
- C is the total number of classes.

The size of S_W is now $d \times d$ i.e a square matrix where d is the number of features or dimensions.

Then, we calculate the overall mean vector by using the formula:

$$\mu = \frac{1}{n} \sum_{i=1}^{C} \sum_{\mathbf{x} \in C_i} \mathbf{x} \tag{12}$$

Where:

- μ is the overall mean vector.
- n is the total number of samples in all classes.

Next step is to calculate the between-class scatter matrix S_B, which represents the spread between different classes. It is computed by summing up the scatter matrices of each class, weighted by their respective class sizes, and then multiplied by the difference between the mean vectors of different classes. Mathematically, it is represented as:

$$S_B = \sum_{c=1}^{C} N_c(\mu_c - \mu)(\mu_c - \mu)^T \tag{13}$$

Where, S_B is the between-class scatter matrix of size $d \times d$

We then calculate the eigenvalues and eigenvectors of matrix $S_W^{-1} S_B$ by using similar equation as equation (4) and equation (5), they provide information about the most discriminative directions in the data. We then sort the eigenvectors by decreasing eigenvalues and pick the top k eigenpairs. Then we construct a matrix W of size $d \times k$ based on the top k most informative eigenpairs where the eigenvectors are kept column-wise. This matrix is then used to transform the matrix X onto the new subspace while maximizing the between-class scatter and minimizing the within-class scatter by the equation:

$$Y = W \cdot X \tag{14}$$

Where, Y is the transformed data matrix of size $n \times k$ and the data has been dimentionally reduced from m dimensions to k dimensions, here k is the number of discriminant components whose maximum value can be $C - 1$. This transformation effectively reduces the dimensionality of the data while preserving class discriminatory information, thus enabling more efficient classification algorithms.

C. INSTRUMENTATION DETAILS

Principal Component Analysis (PCA) was performed using **Jupyter Notebook**, a widely used interactive coding environment. Several key libraries were utilized for data analysis and visualization. The **seaborn** library was employed to load the dataset, and the **numpy** library was used for numerical operations and array manipulation. Data manipulation and visualization were handled using the **pandas** library. The **matplotlib.pyplot** library facilitated the creation of various plots and visualizations. For 3D plotting, the **mpltoolkits.mplot3d** module was imported. Finally, data standardization and scaling were performed using **sklearn.preprocessing**.

D. BLOCK DIAGRAM

The block diagram shown in Figure 1 illustrates the complete process of Principal Component Analysis (PCA). It starts with loading the dataset and conducting preprocessing tasks such as handling missing values and encoding categorical features. Subsequently, the dataset undergoes scaling and normalization using mean and standard deviation. Next, the covariance matrix, eigenvalues, and eigenvectors are computed. The proportion of variance is determined, and a row feature vector matrix P is constructed. Then, the final matrix Y is derived and its plot is visualized. This process is repeated for various combinations of P, and the resulting figures are analyzed to gain insights into the data.

The block diagram shown in Figure 2 illustrates the complete process of Linear Discriminant Analysis (LDA). Similar to PCA, the dataset is first preprocessed by handling missing values, encoding categorical features and scaling and normalizing the data. After that, the mean vectors for each class are computed. Following this, the within-class scatter matrix S_W and the between-class scatter matrix S_B are calculated. The eigenvalues and eigenvectors of the matrix $S_W^{-1}S_B$ are computed, and the eigenvectors corresponding to the largest eigenvalues are selected to form the transformation matrix W. Finally, the dataset is transformed into a new subspace using W, and the transformed data is analyzed to gain insights into class separability.

IV. IMPLEMENTATION DETAILS AND RESULTS

A. PCA ON SYNTHETIC DATASET

First, we sampled twenty points from a standard normal distribution with mean 0 and standard deviation 1, resulting in a matrix of shape 20×2. The plot of these twenty points can be seen in Figure 3.

Next, we sampled a 2×2 matrix from a uniform distribution ranging from 0 to 1, which acted as a transformation matrix aligning the points to a certain axis. Multiplying this transformation matrix with the points matrix yielded our matrix X. The plot of matrix X can be seen in Figure 4.

We then calculated the variance of matrix X along both axes, resulting in 0.4375 and 0.3433 in the horizontal and vertical axes, respectively. The covariance matrix of X was found to be:

$$S_X = \begin{bmatrix} 0.4606 & 0.4003 \\ 0.4003 & 0.3614 \end{bmatrix}$$

Subsequently, we found the eigenvalues and eigenvectors of S_X. The obtained eigenvalues were 0.8144 and 0.0076, and the corresponding eigenvectors were:

$$\begin{bmatrix} 0.7493 & -0.6622 \\ 0.6622 & 0.7493 \end{bmatrix}$$

The eigenvector for each eigenvalue is given column-wise.

The existence of two eigenvalues and eigenvectors suggested the presence of two principal components. The proportion of variance was calculated as 99.0737 and 0.9262.

Using these principal components, we obtained matrix Y, which represents X after a change of basis. The plot of Y is shown in Figure 5. The covariance matrix of Y was obtained as:

$$S_Y = \begin{bmatrix} 8.1444 \times 10^{-1} & -6.4913 \times 10^{-17} \\ -6.4913 \times 10^{-17} & 7.614 \times 10^{-3} \end{bmatrix}$$

When considering only the first principal component, the plot is shown in Figure 6, and when considering only the second principal component, the plot is shown in Figure 7. In both cases, the two-dimensional data has been reduced to a single dimension through dimensionality reduction.

B. PCA ON PALMER PENGUINS DATASET

This dataset comprises 6 attribute features and 344 instances categorized into 3 classes. Initially, we formed a 344×6 matrix from the dataset. After discarding rows with missing values, the dataset reduced to 333 instances, resulting in a 333×6 matrix. Removing two categorical attributes further downsized our data matrix to 333×4. Subsequently, we encoded penguin classes into numerical values (0, 1, and 2) to facilitate data separation for plotting. Standardization and scaling using Python's StandardScaler library transformed our dataset into matrix X, ensuring a mean of zero and a variance of one across all axes.

The covariance matrix of X is found to be:

$$S_X = \begin{bmatrix} 1.003 & -0.2293 & 0.655 & 0.5912 \\ -0.2293 & 1.003 & -0.5795 & -0.4734 \\ 0.655 & -0.5795 & 1.003 & 0.8756 \\ 0.5912 & -0.4734 & 0.8756 & 1.003 \end{bmatrix}$$

Then, we computed the eigenvalues and eigenvectors of S_X. The resulting eigenvalues (2.7536, 0.7804, 0.3697, and 0.1082) and their respective eigenvectors

$$\begin{bmatrix} -0.4537 & 0.6001 & 0.6424 & 0.1451 \\ 0.3990 & 0.7961 & -0.4258 & -0.1599 \\ -0.5768 & 0.0057 & -0.2360 & -0.7819 \\ -0.5496 & 0.0764 & -0.5917 & 0.5846 \end{bmatrix}$$

The eigenvector for each eigenvalue is given column-wise.

led us to discern four principal components. These components explained variances of 68.6338

By selecting the top three principal components, we derived matrix Y, enabling us to visualize our data in three dimensions (starting from figure 8). A covariance matrix S_Y was computed accordingly.

$$S_Y = \begin{bmatrix} 2.7536 \times 10^{0} & 5.3505 \times 10^{-16} & 1.0701 \times 10^{-17} \\ 5.3505 \times 10^{-16} & 7.8046 \times 10^{-1} & 2.6752 \times 10^{-16} \\ 1.0701 \times 10^{-17} & 2.6752 \times 10^{-16} & 3.6975 \times 10^{-1} \end{bmatrix}$$

Further dimensionality reduction was accomplished by considering only two principal components, resulting in two-dimensional visualizations (figure 11). We obtained the corresponding covariance matrix S_Y for this reduced dimensionality.

$$S_Y = \begin{bmatrix} 2.7536 \times 10^{0} & 5.3505 \times 10^{-16} \\ 5.3505 \times 10^{-16} & 7.8046 \times 10^{-1} \end{bmatrix}$$

Finally, we reduced the dataset to one dimension using a single principal component, visualizing the results accordingly (figure 17). The covariance matrix S_Y for this scenario was also determined.

$$S_Y = \begin{bmatrix} 2.7536 \end{bmatrix}$$

In summary, our PCA process successfully reduced the dataset from four dimensions to one dimension, thereby aiding in efficient data visualization.

C. LDA ON THE PALMER PENGUINS DATASET

Following a process similar to PCA, we preprocessed the dataset by removing entries with missing values, discarding categorical attributes, encoding penguin classes numerically, and standardizing and scaling the data using Python's StandardScaler library to obtain our data matrix X.

The mean vectors for each class are as follows:

$$\mu_0 = \begin{bmatrix} -0.9465 \\ 0.6013 \\ -0.7763 \\ -0.6229 \end{bmatrix}, \quad \mu_1 = \begin{bmatrix} 0.8865 \\ 0.6386 \\ -0.3675 \\ -0.5895 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 0.6547 \\ -1.1027 \\ 1.1624 \\ 1.1012 \end{bmatrix}$$

We then computed the within-class scatter matrix S_W as

$$\begin{bmatrix} 97.7148 & 54.3920 & 41.7750 & 59.9247 \\ 54.3920 & 107.7722 & 44.2530 & 67.6186 \\ 41.7750 & 44.2530 & 75.0197 & 53.0239 \\ 59.9247 & 67.6186 & 53.0239 & 108.3952 \end{bmatrix}$$

The overall mean vector is calculated as:

$$\mu = \begin{bmatrix} 3.8407 \times 10^{-16} \\ 6.4012 \times 10^{-16} \\ 2.1337 \times 10^{-16} \\ -1.707 \times 10^{-16} \end{bmatrix}$$

Afterward, we computed the between-class scatter matrix S_B as

$$\begin{bmatrix} 235.2851 & -130.5244 & 175.7058 & 136.3625 \\ -130.5244 & 225.2277 & -236.6577 & -224.7998 \\ 175.7058 & -236.6577 & 257.9802 & 237.6780 \\ 136.3625 & -224.7998 & 237.6780 & 224.6047 \end{bmatrix}$$

Next, we obtained the eigenvalues and eigenvectors for the matrix $S_W^{-1} S_B$. The eigenvalues were 1.503×10^{1}, 2.3362×10^{0}, 4.2041×10^{-16}, and -1.0135×10^{-15} with corresponding eigenvectors given by:

$$\begin{bmatrix} -0.1774 & -0.8566 & 0.0397 & 0.1843 \\ 0.7748 & -0.0077 & -0.7223 & -0.3781 \\ -0.4476 & 0.0751 & -0.0557 & -0.8234 \\ -0.4098 & 0.5103 & -0.6881 & 0.3809 \end{bmatrix}$$

These eigenvectors are provided column-wise.

Using the top k eigenvectors, we derived the transformation matrix W. For our dataset, k was set to 2, given that k can be at most one less than the number of classes, which is 3 in our case. The transformation matrix W is represented as:

$$W = \begin{bmatrix} -0.1774 & -0.8567 \\ 0.7748 & -0.0077 \\ -0.4476 & 0.0751 \\ -0.4098 & 0.5103 \end{bmatrix}$$

Finally, utilizing this transformation matrix, we obtained the final data matrix Y, and its plot is illustrated in Figure 21.

V. DISCUSSION AND ANALYSIS

A. PCA ON RANDOM DATASET

Initially, in the Random Dataset, the data points are randomly scattered, as shown in Figure 3. Upon multiplying by a transformation matrix, the points come closer together, forming a pattern resembling a straight line.

The final step of Principal Component Analysis (PCA) is to find a diagonal matrix S_Y, where the diagonal terms are as large as possible, and the off-diagonal terms are zero. However, in our random dataset, the off-diagonal terms were not exactly zero but very close, around 10^{-16}. This small deviation is likely due to computational limitations, as the laptop could not achieve higher precision.

After applying PCA, using two principal components, the data points were transformed to align with a new coordinate system, where the axes are the principal components. This transformation is like to rotating the data points with a transformation matrix. When using only one of the principal components, the dimensionality was reduced to one, and the data points aligned in a straight line.

B. PCA ON PALMER PENGUIN DATASET

In the Pamler Penguin dataset, initially the multidimensional nature of the Palmer Penguins dataset posed visualization challenges due to its four-dimensional complexity. However, employing Principal Component Analysis (PCA) facilitated dimensionality reduction, enabling visualization in 1, 2, and 3 dimensions.

Utilizing only one principal component failed to provide much information as shown in figure 17 as all the data points collapsed in a single line and the nothing insightful or distinguishable pattern can be deduced. When using two principal components data understanding is enhanced and more information can be gained. Figure 11 shows the usage of principal component PC0 and PC1 and figure 12 shows the usage of principal component PC0 and PC2. From comparing these representations PC0 and PC2 showed cleared data separation compared to PC0 and PC1. For other combinations there is not many useful information as the data is jumbled and cannot be separated.

PC0, capturing the highest proportion of variance (68.63%), represents the primary direction of variability in the dataset. PC1, with a proportion of variance of 19.45%, accounts for the secondary direction of variability, while PC2 (9.22%) and PC3 (2.70%) capture subsequent, diminishing proportions of variance. Despite PC1 having a substantial proportion of variance, its alignment with the factors driving data separation might not be as strong as PC2, which captures a significant but distinct aspect of the dataset's variability. Therefore, the combination of PC0 and PC2 might better align with the critical factors for data separation, resulting in clearer distinctions between data classes as compared to PC0 and PC1.

Expanding to three principal components as shown in figures starting from figure 8 where principal components PC0, PC1, PC2 are used, results in more effective separation of penguin classes. In figure 9, where PC0, PC1 and PC3 are used, the data for Gentoo penguins is separated, but Adelie and Chinstrap penguins cannot be distinguished. When using PC1, PC2, and PC3, the data cannot be separated effectively, likely because PC0 captures the most significant variance (68.63%). This dominant variance is critical for distinguishing the major structure and separability within the dataset. By excluding PC0, the principal component space loses this primary source of variance, making it difficult to effectively separate the data.

C. LDA ON PALMER PENGUIN DATASET

In contrast to PCA, LDA operates with the goal of maximizing class separability rather than capturing variance. When applied to the Palmer Penguin dataset, LDA offers insights into the underlying structure of the data, specially in distinguishing between different penguin species.

Initially, the multidimensional nature of the Palmer Penguins dataset posed visualization challenges due to its four-dimensional complexity. However, LDA facilitated dimensionality reduction, enabling visualization in lower dimensions. Utilizing LDA, we derived a transformation matrix W to project the data onto a lower-dimensional subspace while maximizing between-class separability and minimizing within-class scatter. The resulting transformation provided clearer distinctions between penguin classes.

The eigenvalues of $S_W^{-1}S_B$ reveal the proportion of variance captured by each discriminant component. The top eigenvalues correspond to the most significant sources of class separability and the corresponding eigenvectors represented the directions in which the data exhibits the greatest separation between classes. By aligning the data along these directions, we maximized the discriminative power of the transformed dataset and data is easily separated as shown in the figure 21.

D. COMPARISION BETWEEN PCA AND LDA

Principal component analysis (PCA) and Linear discriminant analysis (LDA) are two widely-used dimensionality reduction techniques applied in various data analysis tasks. Despite both involving linear transformations through eigenvalue decomposition, the underlying objectives and the way they achieve dimensionality reduction are fundamentally different for both PCA and LDA. The main objective of PCA is to maximize the variance of data, seeking to capture the directions of maximum variance in the feature space. The mathematical procedure used by PCA for dimensionality reduction is eigenvalue decomposition. PCA reduces dimensionality by selecting the principal components. On the other hand, the main objective of LDA is to maximize separation between the classes in the dataset. The mathematical procedure used by LDA for dimensionality reduction is by maximizing between-class scatter matrix and minimizing within-class scatter matrix. LDA reduces dimensionality by finding a linear transformation that maximizes the separation between classes in the dataset. LDA achieves this by projecting the data onto a lower-dimensional space where the classes are well-separated. PCA is used for high-dimensional datasets and is suitable when the relationships between variables are mostly linear. LDA, on the other hand, is used for classification tasks and supervised dimensionality reduction, performing well when each class follows a normal distribution.

In our analysis of the Palmer Penguin dataset, we used both PCA and LDA to reduce the number of features. We noticed that PCA emphasizes the spread of data, keeping the most important patterns. On the other hand, LDA aims to make the different classes more distinct, creating a simpler representation. Figure 11 illustrates the distribution of data points in the principal components obtained through PCA, highlighting the variance-maximizing axes in a two-dimensional space. In contrast, Figure 21 demonstrates the projection of data points onto a lower-dimensional space using LDA, emphasizing the enhanced separation between classes achieved by this technique. We can observe that the dataset are well seperated on the basis of class using LDA in two-dimensional space than using PCA in two largest principal components.

VI. CONCLUSION

In conclusion, the exploration of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) has yielded insightful observations and comparisons.

PCA demonstrated its effectiveness in feature extraction and dimensionality reduction, capturing underlying structures across diverse datasets, including synthetic data and real-world examples like the Palmer Penguins dataset. Through various visualizations, PCA showcased its ability to condense information while preserving discriminatory details, thus enhancing data visualization and potential classification tasks.

While both PCA and LDA have their respective strengths and limitations, their applications offer advantages in data analysis. PCA excels in capturing overall variance and revealing underlying structures within datasets, making it very useful for exploratory analysis. On the other hand, LDA is good at optimizing class separability, which benefits tasks involving classification.

Exploration and experimentation with these techniques, improve our understanding of data analysis and pattern recognition. By using the strengths of PCA and LDA, researchers and practitioners can find new insights in various domains, ultimately enhancing the ability to extract meaningful information from complex datasets.

APPENDIX

A. BLOCK DIAGRAM

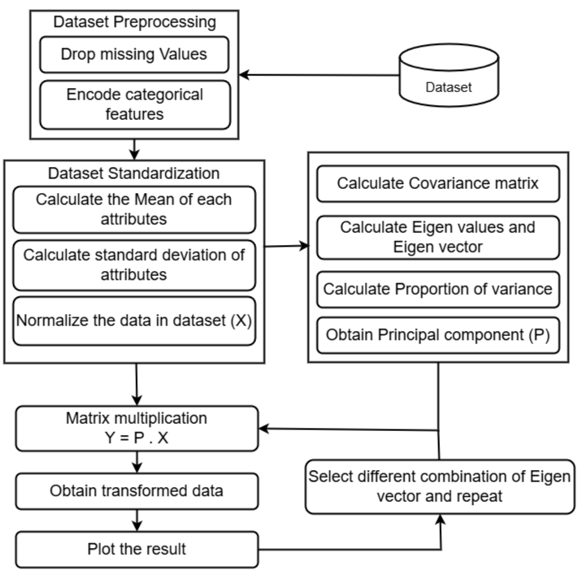

FIGURE 1. Block diagram of PCA

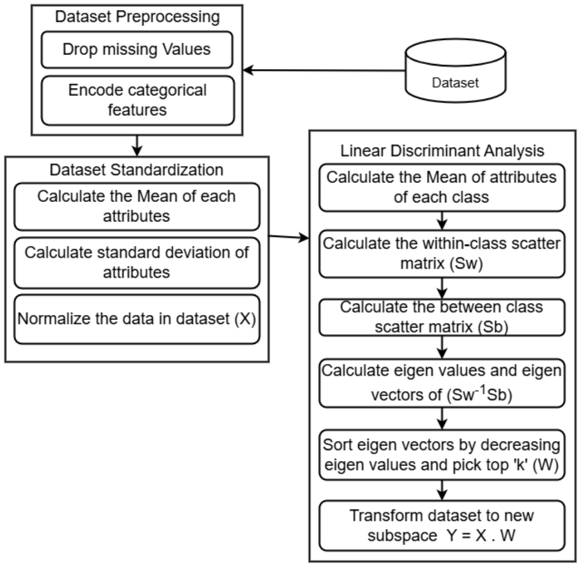

FIGURE 2. Block diagram of LDA

B. PLOTS FOR PCA ON SYNTHETIC DATASET

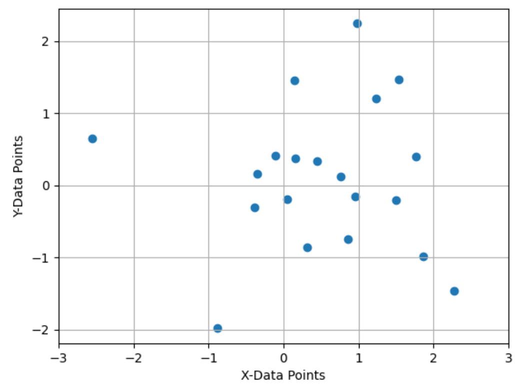

FIGURE 3. Data with standard normal distribution

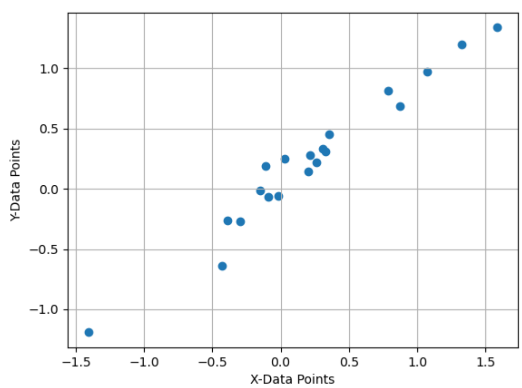

FIGURE 4. Distribution of matrix *X*

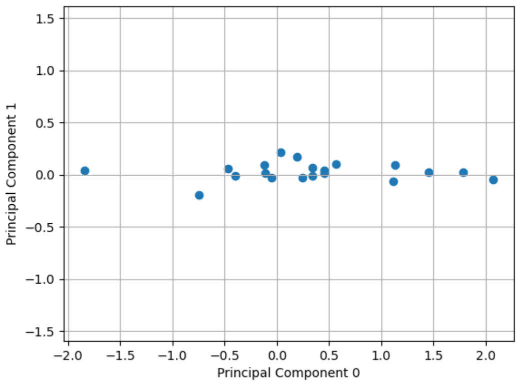

FIGURE 5. **Plot of matrix *Y* taking both principal components**

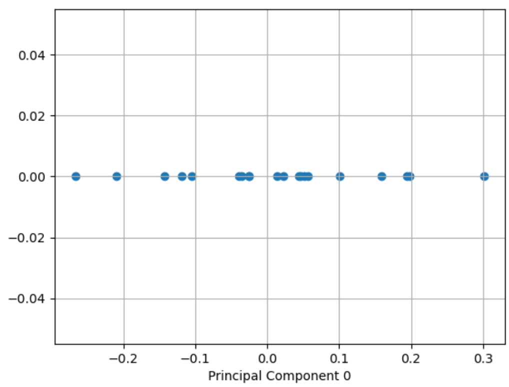

FIGURE 6. **Plot of matrix *Y* taking PC0**

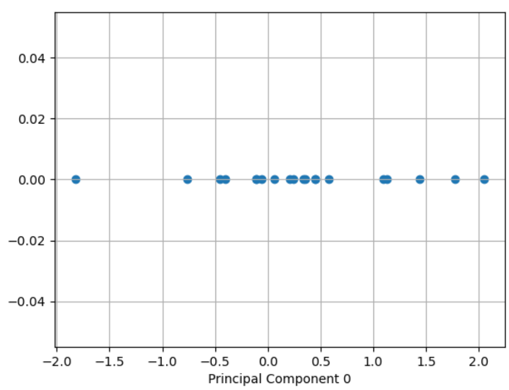

FIGURE 7. **Plot of matrix *Y* taking PC1**

C. PLOTS FOR PCA ON PALMER PENGUINS DATASET

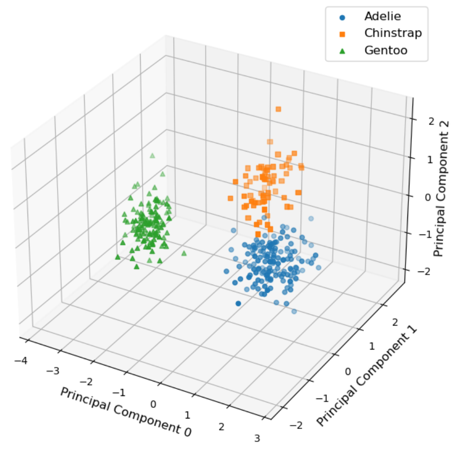

FIGURE 8. **3D plot of Palmer Penguins dataset using PC0, PC1 and PC2**

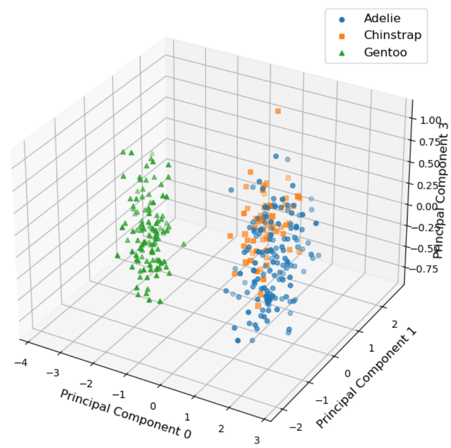

FIGURE 9. **3D plot of Palmer Penguins dataset using PC0, PC1 and PC3**

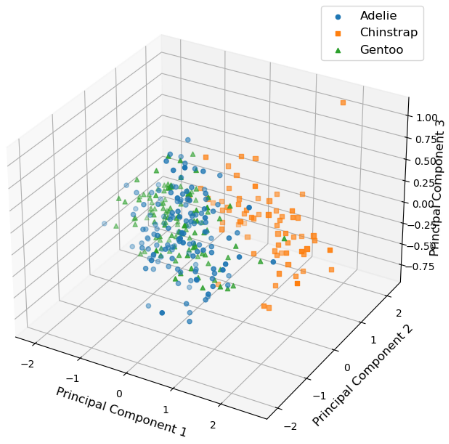

FIGURE 10. **3D plot of Palmer Penguins dataset using PC1, PC2 and PC3**

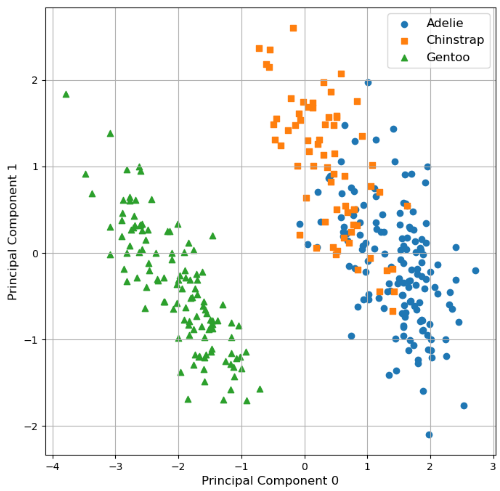

FIGURE 11. **2D plot of Palmer Penguins dataset using PC0 and PC1**

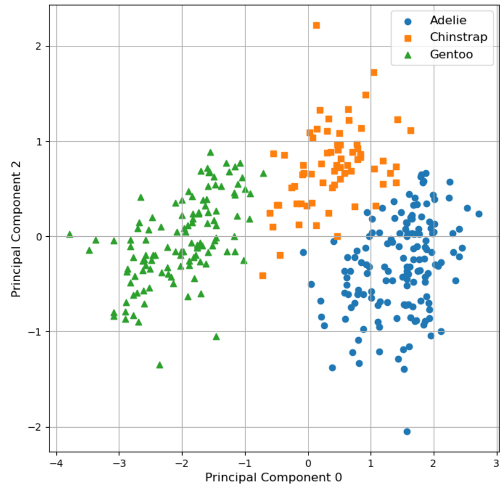

FIGURE 12. **2D plot of Palmer Penguins dataset using PC0 and PC2**

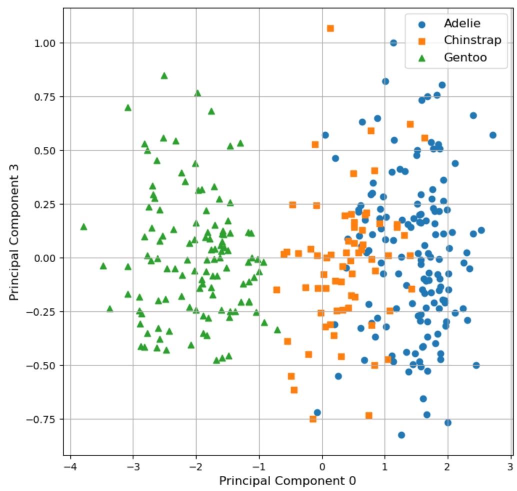

FIGURE 13. **2D plot of Palmer Penguins dataset using PC0 and PC3**

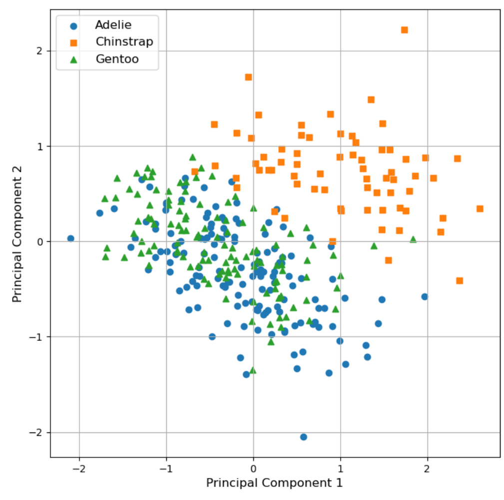

FIGURE 14. 2D plot of Palmer Penguins dataset using PC1 and PC2

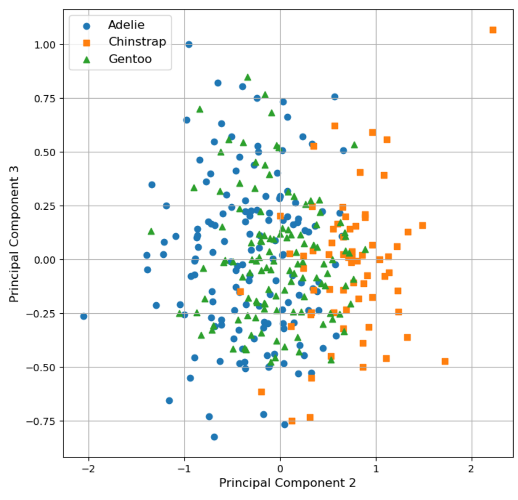

FIGURE 16. 2D plot of Palmer Penguins dataset using PC2 and PC3

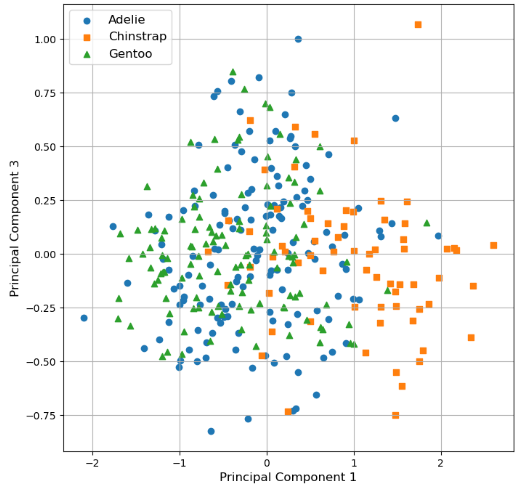

FIGURE 15. 2D plot of Palmer Penguins dataset using PC1 and PC3

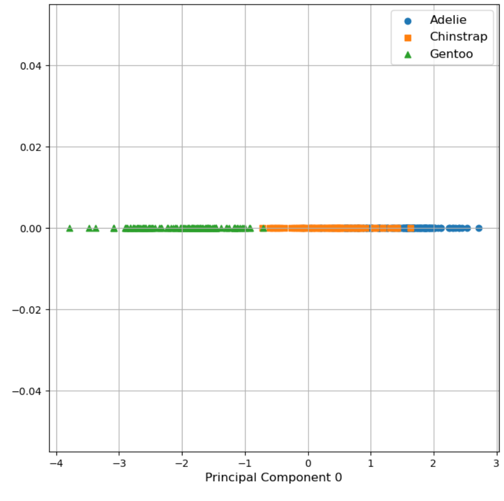

FIGURE 17. 1D plot of Palmer Penguins dataset using PC0

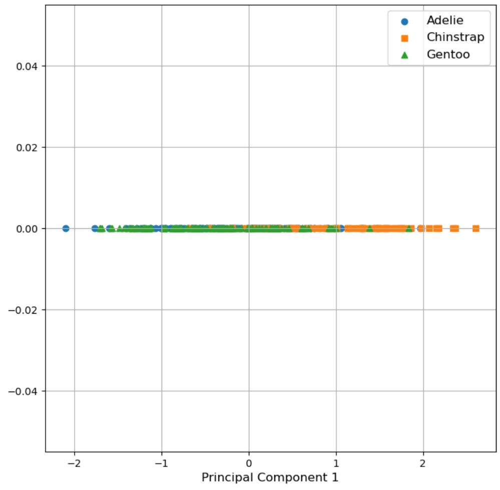

FIGURE 18. **1D plot of Palmer Penguins dataset using PC1**

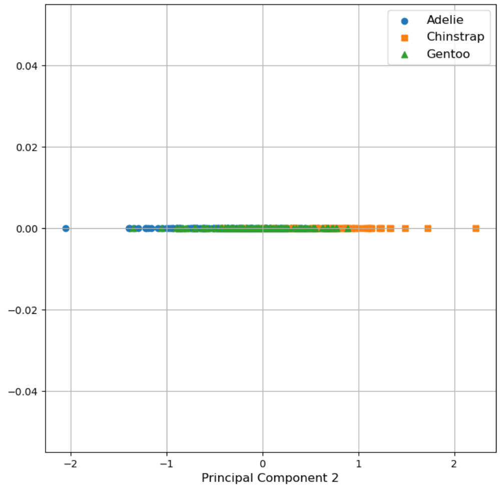

FIGURE 19. **1D plot of Palmer Penguins dataset using PC2**

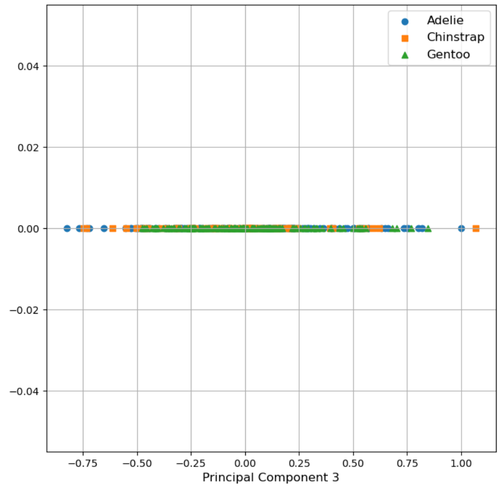

FIGURE 20. **1D plot of Palmer Penguins dataset using PC3**

D. PLOT FOR LDA ON PALMER PENGUINS DATASET

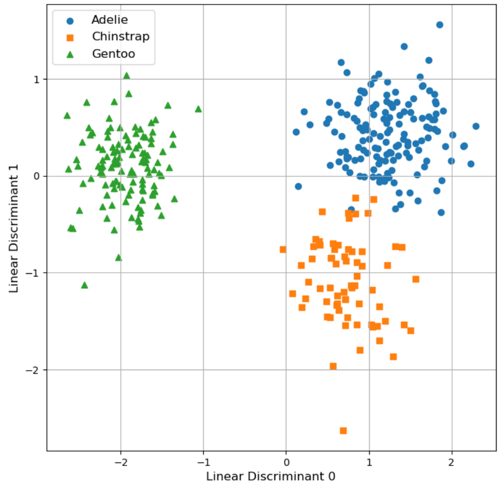

FIGURE 21. **2D plot of Palmer Penguins dataset using LD0 and LD1**

REFERENCES

[1] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY, USA: Springer, 2002

[2] Shlens, J., *"A Tutorial on Principal Component Analysis,"* 2014. [Online]. Available: https://arxiv.org/abs/1404.1100.

[3] F. Gewers et al., *"Principal Component Analysis: A Natural Approach to Data Exploration,"* ACM Computing Surveys, vol. 54, pp. 1-34, 2021.

[4] I. T. Jolliffe and J. Cadima, *"Principal component analysis: a review and recent developments,"* Phil. Trans. R. Soc. A., vol. 374, 2016, Article ID 20150202. [Online]. Available: http://doi.org/10.1098/rsta.2015.0202.

[5] H. Zhao, Z. Lai, H. Leung, and X. Zhang, *"Linear Discriminant Analysis,"* in *Feature Learning and Understanding, Information Fusion and Data Science*, Springer, Cham, 2020, pp. 63-75. [Online]. Available: https://doi.org/10.1007/978-3-030-40794-0_5.

[6] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, *"Linear discriminant analysis: A detailed tutorial,"* AI Communications, vol. 30, pp. 169-190, 2017.

[7] S. Balakrishnama and Aravind Ganapathiraju, *"Linear Discriminant Analysis—A Brief Tutorial,"* IEEE Transactions on Neural Networks, vol. 11, 1998.

NIMESH G. PRADHAN is currently pursuing a Bachelor's degree in Electronics, Communication, and Information Engineering at Thapathali Campus. He is currently in the final year of his degree. His interests lie in the fields of Data Mining, Machine Learning, and Deep Learning.

RAMAN BHATTARAI is currently pursuing a Bachelor's degree in Electronics, Communication, and Information Engineering at Thapathali Campus. He is currently in the final year of his degree. His interests lie in the fields of Data Mining, Computer Vision, and Deep Learning.