

| **Title: Implementation of Principal Component Analysis** |
| --- |

**Objective:** To implement Principal Component Analysis.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

<https://numpy.org/doc/stable/reference/generated/numpy.argsort.html>

<https://math.libretexts.org/Bookshelves/Linear_Algebra/A_First_Course_in_Linear_Algebra_(Kuttler)/07%3A_Spectral_Theory/7.01%3A_Eigenvalues_and_Eigenvectors_of_a_Matrix>

<https://www.javatpoint.com/principal-component-analysis>

**Theory of Principal Component Analysis**

Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components. It is one of the popular tools that is used for exploratory data analysis and predictive modeling. It is a technique to draw strong patterns from the given dataset by reducing the variances.

PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are image processing, movie recommendation system, optimizing the power allocation in various communication channels. It is a feature extraction technique, so it contains the important variables and drops the least important variable.

**Dataset details used in the Experiment**

Sample dataset

| Fearture | Exp1 | Exp2 | Exp3 | Exp4 |
| --- | --- | --- | --- | --- |
| x | 4 | 8 | 13 | 7 |
| y | 11 | 4 | 5 | 14 |

**Explanation of API/Tool used for implementation**

Numpy - used to calculate eigenvalues and eigenvectors from covariance matrix.

**Source Code:** Attached in separate file

**Results/Output**

| Feature | Exp1 | Exp2 | Exp3 | Exp4 |
| --- | --- | --- | --- | --- |
| Principle component 1 | 0.15393283 | 3.73612869 | 0.11892802 | -4.00898954 |

**Conclusion:** Thus we have performed principle component analysis. We have done the analysis by using python and numpy without using any external libraries.