# PRINCIPAL COMPONENT ANALYSIS AND ITS EXTENSIONS IN MACHINE LEARNING: THEORY, APPLICATIONS AND ADVANCES

A Dissertation

Submitted in partial fulfilment of the requirement for the degree of

## MASTER OF SCIENCE

in

## APPLIED MATHEMATICS

By:

**JASMINE**

**23/MSCMAT/24**

**Under the supervision of**

**Ms. TRASHA GUPTA**

**Department of Applied Mathematics**

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Shahabad Daulatpur, Bawana Road, Delhi-110042, India

June, 2025

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

# CANDIDATE’S DECLARATION

I, **Jasmine**, hereby certify that the work is being presented as the Major Project in the thesis entitled **" Principal Component Analysis And Its Extensions In Machine Learning: Theory, Applications and Advances"** in partial fulfilment of the requirement for the award of the Degree of Master of Science in Applied Mathematics and submitted to the Department of Applied Mathematics, Delhi Technological University, Delhi is an authentic record of my work, carried out during the period from January 2025 to May 2025 under the supervision of **Ms. Trasha Gupta**.

I have not submitted the matter presented in the report for the award of any other degree of this or any other institute/University.

Place: Delhi

**Jasmine 23/MSCMAT/24**

Date:

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

## CERTIFICATE BY THE SUPERVISOR

Certified that **Jasmine(23/MSCMAT/24)** has carried out her research work presented in this thesis entitled " **Principal Component Analysis And Its Extensions In Machine Learning: Theory, Applications and Advances** " for the award of Degree of Master of Science in Applied Mathematics and submitted to the Department of Applied Mathematics, Delhi Technological University, Delhi under my supervision. This thesis embodies results of original work, and studies are carried out by the student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**Ms. Trasha Gupta**
Department of Applied Mathematics
Delhi Technological University

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

## PLAGIARISM VERIFICATION

Title of the thesis

________________________________________________________________________

________________________________________________________________________

Total Pages _______

Name of the Scholar __________________________

Supervisor

_______________________________

Department __________________________________________

This is to report that the above thesis was scanned for similarity detection. The process and outcome are given below:

Software used: ___________________ Similarity Index _____

Total Word Count: ________

Date:

**Candidate's signature** **Signature of Supervisor**

# ABSTRACT

In today’s vast and complex data landscapes, every model or prediction we make is influenced by a multitude of factors, often referred to as dimensions in the context of machine learning. In such cases, reducing the dimensionality of the data can significantly simplify the model without losing important information. By reducing the dimensions, we can focus only on the most crucial factors, making the model more efficient and accurate. This is where PCA comes in – it helps reduce dimensionality while retaining the most important features, making models more efficient, easier to interpret and computationally feasible. Therefore, PCA becomes an essential tool in handling high-dimensional data. To give learners an extensive understanding of PCA function in today's machine learning, this dissertation studies the mathematical basis, expansions, and applications of the method. Different properties of PCA and its variants like sparse PCA, kernel PCA, incremental PCA and robust PCA are highlighted with detailed derivations and graphical interpretations. We have Applied PCA to different types of datasets across multiple domains, including bioinformatics, genome analysis or computer vision to evaluate its practical effectiveness. In recent years, the combinations with PCA has proved or solved many problems. And, these advancements like PCA with cluster analysis or PCA combined with discriminant analysis help in reducing dimension obviously, reducing noise and better feature extraction since data in this huge space is very complex. While PCA has various advantages, it is not always the perfect solution for every problem. Researchers often face several challenges when applying PCA in experiments. One major limitation is its assumption of linearity. Additionally, PCA is highly sensitive to noisy or irrelevant features, which can lead to misleading results. Another challenge is the difficulty in interpretability of new components, as they are linear combinations of the original variables rather than direct meaningful features.

# ACKNOWLEDGEMENT

At the time of submission of my M.Sc. Dissertation, I express my deepest gratitude to the divine God for providing me with the insight, strength, and perseverance to complete this journey. The successful completion of this project would not have been possible without the support, encouragement, and guidance of many individuals, to whom I am sincerely thankful.

First and foremost, I am profoundly grateful to Delhi Technological University for providing me with the platform and resources to pursue my research work.

I wish to extend my heartfelt thanks to my supervisor, Ms. Trasha Gupta, Department of Applied Mathematics, for offering me the opportunity to work under his esteemed guidance. His invaluable mentorship, expertise, continuous support, and insightful suggestions have been pivotal to the successful completion of this work.

**Jasmine**
**23/MSCMAT/24**

# List of Figures

3.3 Robust PCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

3.5 Incremental PCA . . . . . . . . . . . . . . . . . . . . . . . . . 19

5.1 Challenges . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

5.2 Limitations of PCA in non linear transformation . 27

5.3.1 PCA with Clustering . . . . . . . . . . . . . . . . . . . . 29

5.3.2 PCA with Discriminant Analysis . . . . . . . . . . 30

# CONTENTS

Candidate's Declaration . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ii

Certificate by the Supervisor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iii

Plagiarism Verification . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iv

Abstract . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . v

Acknowledgement . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . vi

List of Figures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . vii

**CHAPTER 1**

1.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.2 Problem Statement . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

1.3 Research purpose . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.4 Importance of the study . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

**CHAPTER 2**

2.1 Maths intuition behind traditional PCA . . . . . . . . . . . . . . . . . . . 5

2.1.1 Maximizing Variance . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2.1.2 Minimum Squared Reconstructed Error . . . . . . . . . . . . . 8

2.2 Eigen decomposition vs Singular value decomposition . . . . . . . 10

**CHAPTER 3**

3.1 Variants of PCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

3.2 Kernel PCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

3.3 Robust PCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

3.4 Sparse PCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

3.5 Incremental PCA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

**CHAPTER 4**

4.1 PCA in bioinformatics and genome analysis . . . . . . . . . . . . . . 20

4.2 PCA in computer vision and image processing . . . . . . . . . . . . 22

**CHAPTER 5**

5.1 Challenges in handling high dimensional data . . . . . . . . . . . . . 24

5.2 Limitations of PCA in non-linear transformation . . . . . . . . . . 26

5.3 Possible advancements in hybrid approaches . . . . . . . . . . . . . 28

5.3.1 PCA combined with clustering . . . . . . . . . . . . . . . . . . 28

5.3.2 PCA combined with discriminant analysis . . . . . . . . . 30

REFERENCES

# CHAPTER 1

## 1.1 INTRODUCTION

Let us begin by discussing what motivates the concept of Principal Component Analysis (PCA). Why do we need PCA? Given that there are numerous other algorithms available for data analysis, one might wonder what makes PCA stand out. This is exactly what we will explore. In today's vast and complex data landscapes, every model or prediction we make is influenced by a multitude of factors, often referred to as dimensions in the context of machine learning. As the number of dimensions increases, several challenges emerge. These challenges manifest as computational difficulties, problems in data analysis, and issues related to storage and memory. The more dimensions we deal with, the more complex and resource-intensive the problem becomes. However, sometimes the data is much simpler than it initially appears. Often, the key to understanding the data lies not in every single feature, but only in a subset of them. In such cases, reducing the dimensionality of the data can significantly simplify the model without losing important information.

For instance, consider a scenario where we want to predict whether an animal is a cat or a dog, simply by touching it. In this case, the colour feature is irrelevant, and reducing the dimensionality by removing unnecessary features like colour will streamline the model.

Similarly, let's consider an image of the digit '7' in a 52x52 pixel format (which is 2,704 - dimensional space because 52 times 52 equals to 2,704). [2] If the image is expanded to 64x64 pixels, it becomes a much larger image with far more data. However, the majority of these additional pixels may not significantly affect the image's interpretation. Only a few key features within this larger space are truly important for recognizing the digit. By reducing the dimensions in this way, we can focus only on the most crucial factors, making the model more efficient and accurate. This is where PCA comes in – it helps reduce dimensionality while retaining the most important features, making

models more efficient, easier to interpret, and computationally feasible. Therefore, PCA becomes an essential tool in handling high-dimensional data.

So, now we can conclude that PCA is used for applications like reducing the dimension, visualizing the data, or selecting the features. We have to ensure that data will have the as much variance as possible. By doing this , the most important patterns or data will be there and duplications will be removed. PCA helps in simplifying complicated datasets and improving efficiency.

## 1.2 PROBLEM STATEMENT

Despite its usefulness, PCA also has several challenges:

- **Linear Relationships** – PCA is helpful in lot of cases like to identify data variance, but the problem is it consider data to be linear always. Therefore, when dataset is complex or non-linear, it fails to provide any help as it only depends on linear transformation.

- **Unseen Data and Outliers** –PCA learns the complexity of data set very well therefore whenever there is any unseen data or outlier, its prediction differs significantly. Predictions may become unstable as a result, which reduces its reliability for real-world applications where data is continuously changing.

- **Cost Problem** –PCA demands high cost while performing any computations which is required in PCA such as any simplifications needed in solving matrices. Without proper optimization, processing very high-dimensional data becomes difficult as the number of features increases since it takes more time and memory.

Various kinds of PCA variants and extensions have been introduced to tackle these issues, each of which is used to handle one of the problem. For bettering PCA's performance and flexibility in a variety of situations, scientists have studied and suggested several modifications. For example, one of the biggest problem of PCA is that it considers the relationship to be linear always, so to handle the complex datasets various advanced techniques have been made. These changes guarantee that PCA remains to be an effective technique even when traditional linear transformations are unable to provide useful findings.

These improved PCA versions make it more accurate, flexible, and appropriate for a greater variety of real-world applications by improving the fundamental methods. This will include:

1. Kernel PCA (KPCA): This variation of PCA is presented only to handle the problem of linearity in traditional PCA. Standard PCA might not give the accurate result when applied to non linear or on complex datasets. This same kind of technique is used in support vector machine, so the idea behind this is same. by applying kernel PCA, it mainly reduce the dimension of the non-linear dataset.[1]

2. Sparse PCA (SPCA) : As evident from the name “sparse” means few or very less. In the context of machine learning, Sparse PCA is presented to maintain the interpretability. This kind of PCA works on this problem by selection small number of principal components, so that accuracy will remain intact and the relevant information will not get lost. Unlike traditional PCA, which creates

full principal components that may involve contributions from all original features. So by this way, we can easily analyse or interpret the results.

3. Robust PCA (RPCA): Traditional PCA's sensitivity to noise and outliers is one of its major drawbacks. To solve this problem, Robust PCA (RPCA) is used, which efficiently handles misclassified data while maintaining the accuracy and significance of the principal components. Data can be made more stable and robust by breaking it down into low rank and sparse components.[2]

4. Incremental PCA (IPCA): Though most of the problems of PCA are solved such as interpretability, non-linearity etc. but one of the major challenge is left which is when data is very large. Traditional PCA becomes extremely costly and demands a large amount of memory and processing power when working with high-dimensional data. Due to limits in handling large-scale computations, this can have an influence on accuracy as well as efficiency. Incremental PCA (IPCA) was created to address this. IPCA divides the data into smaller quantities and applies an individual fit to each batch instead of analysing the complete dataset all at once.

## 1.3 RESEARCH PURPOSE

To give learners an extensive understanding of PCA function in today's machine learning, this dissertation studies the mathematical basis, expansions, and applications of the method. In particular, it seeks to:

- Different properties of PCA and its variants like sparse PCA, kernel PCA, incremental PCA and robust PCA.
- To learn which type of PCA suits best in different kind of conditions or scenarios.
- Examine the extensions of PCA and its connections with eigenvalues and eigenvectors, providing deeper insights into its theoretical framework.
- Apply PCA to different types of datasets across multiple domains, including image processing, finance, healthcare, and anomaly detection, to evaluate its practical effectiveness.

## 1.4 IMPORTANCE OF THE STUDY

Large datasets exist in a huge feature space, making it difficult to interpret the results effectively. As the number of dimensions increases, extracting meaningful insights becomes more complex. PCA plays a crucial role in addressing this challenge by reducing dimensionality while minimizing information loss, allowing for better analysis and interpretation of data**.** Dimensionality reduction is the major reason why we have used this technique as it will preserve the important information. It is helpful in various sectors such as healthcare, finance etc. Various types of PCA will reduce the problem whether it is sensitivity towards outliers or computational cost, every problem is resolved. Therefore, to know more, to learn about this study is very important as there might be some drawback. Let's dive into the world of PCA, its variants and advances in machine learning. By exploring more in different types of PCA such robust, incremental, sparse or kernel PCA, we can know more about which type will work best on which dataset.

# CHAPTER 2

## 2.1 MATHS INTUITION BEHIND TRADITIONAL PCA

To truly learn about the mathematics behind Principal Component Analysis (PCA), it is essential to have a strong understanding of statistics and linear algebra. These mathematical foundations help in deriving the core algorithms of PCA, which essentially revolve around two key optimization approaches:

1. Maximizing the variance of the projected data points.[2]

2. Minimizing the squared reconstruction error.[2]

### 2.1.1 We will begin by exploring the first algorithm, which is based on maximizing variance.

At this point, an important question arises: Why do we focus on variance rather than any other statistical measure? The answer is quite easy—we can use variance because it has squared term in its formula, which will be helpful when we differentiate the term. Because of this feature, it works well for optimization issues and enables us to calculate principal components quickly and effectively using methods based on calculus. To better understand why maximizing variance is crucial, let's consider a simple real-world example. Imagine an assembly of students standing in a row. A teacher wants to count the total number of students in the assembly. Now, if the teacher stands at the side of the row and counts, it becomes much easier to see all the students at once because the spread is maximized. However, if the teacher stands directly in front of the row, the students appear to be stacked one behind the other, making it difficult to differentiate between them. This example clearly demonstrates why maximizing variance is important in PCA. By spreading out the data as much as possible in a new coordinate system, the most important patterns and variations in the data are efficiently captured thanks to PCA. Better dimensionality reduction and feature extraction result from the transformation being more relevant as the spread increases. Therefore, maximizing variance is a key idea in the mathematical formulation of PCA since it enables it to keep the most information from the original dataset while lowering dimensionality.

Let's consider we have N data points denoted as $\{a_n\}$ where n = 1,2 ….., N and each data point has M dimensions (features). To make things simpler, we assume a 2-d dataset and aim to reduce it to 1-d using PCA. PCA finds a new coordinate system, where each axis represents a Principal Component (PC). These principal components capture the maximum variance in the data and are denoted as:

- PC1, PC2, ..., PC'M (depending on the number of dimensions).

Key Constraints of Principal Components:

1. The number of Principal Components (PCs) is always less than or equal to the number of original features. That is, if the data has M features, then the number of PCs will be at most M.
2. All Principal Components are orthogonal (perpendicular) to each other, meaning they form a new set of basis vectors that are uncorrelated.

Now we will first compute the mean:

$$\bar{a} = \frac{1}{N}\sum_{n=1}^{N} a_n$$

This mean-centered transformation helps in simplifying the computations, making it easier to find the principal components that best represent the data.

We shall choose a unit vector such that $v^T v = 1$, as we are only interested in direction.

$$\text{Variance} = \frac{1}{N}\sum_{n=1}^{N}\{ v^T a_n - v^T \bar{a}\} = v^T X v$$

Where X is co-variance matrix, defined by:

$$X = \frac{1}{N}\sum_{n=1}^{N}(a_n - \bar{\text{a}})(a_n - \bar{a})^T$$

We now maximize the variance with respect to $v$. Now, from the properties of matrices we know that

$$X\, v = \lambda v$$

Where $v$ is the eigen vector of X.

$$v^T X v = \lambda$$

The eigen vector corresponding to largest eigen value $\lambda$ is known as principal component or referred as PC1. Python provides an easy way to perform this computation using NumPy’s linalg.eig() function. This function allows us to efficiently extract eigenvalues and eigen vectors.

Summarizing the PCA , it includes steps:

Standardisation:  Normalize the dataset to ensure all features contribute equally.

Compute Covariance Matrix – Captures relationships between different features.

Find Eigenvalues & Eigenvectors – Eigenvectors define new axes, eigenvalues indicate variance captured.

Select Principal Components – Choose eigenvectors with the highest eigenvalues as PC.

Project Data onto PCs – Transform data into the new lower-dimensional space.

### 2.1.2 PCA VIA MINIMUM SQUARED RECONSTRUCTED ERROR

Previously, we studied the variance maximization approach, where PCA finds the principal components that capture the maximum variance in the dataset. Now, we will explore another mathematical perspective on PCA: minimizing the squared reconstruction error. This approach gives us the geometric representation of PCA as finding a subspace without much loss of information. Now we will understand the maths behind this approach.

We want to minimize $\frac{1}{N}\sum_{n=1}^{N}\|a_n - \tilde{a}_n\|^2$

Now we will introduce orthonormal basis, such that $v_i^T v_j$ is 1 when i = j and otherwise 0.

New basis will be,

$$a_n = \sum_{i=1}^{M} b_{ni}\ v_i$$

Multiply $v_j^T$ on both sides,

$$b_{ni} = a_n^T v_i$$

Implies $a_n = \sum_{i=1}^{M}(a_n^T v_i)\ v_i$

For reconstruction, $\widetilde{a_n} = \sum_{i=1}^{m}(a_n^T v_i)\ v_i + \sum_{i=m}^{M} z_i\ v_i$

$a_n$-$\widetilde{a_n} = \sum_{i=1}^{M}(a_n^T v_i)\ v_i - \sum_{i=1}^{m}(a_n^T v_i)\ v_i - \sum_{i=m}^{M} z_i\ v_i$

$= \sum_{i=m}^{M}(a_n^T v_i)\ v_i - \sum_{i=m}^{M} z_i\ v_i$

$= \sum_{i=m}^{M}\ (a_n^T v_i)\ v_i - z_i v_i$

$$\frac{1}{N}\sum_{n=1}^{N}\|a_n - \tilde{a}_n\|^2 = \frac{1}{N}\sum_{n=1}^{N}\left\|\sum_{i=m}^{M} (a_n^T v_i)\ v_i\ -\ z_i v_i\right\|^2$$

Let $z_i = \bar{a}^T v_i$, we are left with

$= \frac{1}{N}\sum_{n=1}^{N}\|((a_n - \widetilde{a_n})v_i)v_i\|^2$

Expand the square using the property $\|Z\|^2 = ZZ^T$

After simplification, we got

$\sum_{i=m}^{M} v_i\, X {v_i}^T$ , where X = covariance matrix.

It started to look lot like what we have studied previously, where

${v_i}^T v_i$ = 1,

Therefore, we can conclude that minimum squared reconstructed error is equal to maximum projected variance.

Both approaches in PCA—variance maximization and reconstruction error minimization—serve different purposes depending on the nature of the data and the objective. maximizing variance is the recommended approach when working with strongly correlated features since it minimizes redundancy while maintaining the most important patterns in the data. This method is mainly useful for feature selection and improving performance by ensuring that the resultant components are uncorrelated and retain maximum information. However, the reconstruction error minimization strategy works better when dimensionality reduction is the main objective. This is particularly important in cases where preserving reconstruction quality is essential, such as data compression, image processing, and managing high-dimensional information.

## 2.2 EIGEN DECOMPOSITION VS SINGULAR VALUE DECOMPOSITION (SVD)

The maths intuition behind this is that basic matrix factorization approach in linear algebra is called singular value decomposition (SVD), which allows every given matrix to be broken down into the product of three unique matrices. To put it simply, a matrix can be factorized into components that reveal significant elements, much like we can express a number as the product of its prime factors. This decomposition is especially helpful for data compression, noise filtering, and dimensionality reduction.

Real-Life Example of SVD:

To understand its practical application, consider an underground pipe through which water flows. Suppose we want to calculate the volume of the pipe, which primarily depends on the fraction of water flow. However, there are additional factors such as speed, temperature, pressure, and other physical parameters that make the problem more complex. Since not all of these dimensions contribute equally to our primary objective, we can apply SVD to reduce the number of variables while still retaining the most crucial information. This allows us to focus only on the significant components, simplifying the calculations without losing critical information.

A key **advantage** of SVD over traditional Eigen Decomposition is that we can use this to any matrix, whether square or rectangular. Eigen decomposition, on the other hand, is only for square matrices. Because of this, SVD is a more flexible and reliable method that can be used in a wider range of fields, including recommendation systems, signal processing, and machine learning.

Mathematical Representation of SVD :

For any given matrix B(size is p×n) , we can decompose it as

$$B = U\sum V^T$$

U is an p×p **unitary (orthogonal) matrix** containing left singular vectors.

Σ (Sigma) is p×n **diagonal matrix** containing singular values, which represent the importance of each component.

$V^T$ is n×n **unitary (orthogonal) matrix** containing right singular vectors.

When there are large datasets, simplifying the decomposition might be expensive and extremely time consuming. In such cases, we try to ignore the less important values and focus just on the important singular values which are likely to more helpful in future. This process is referred as economy singular value decomposition. This is particularly useful in dimensionality reduction, where we ignore low-importance singular values to improve efficiency without significantly affecting accuracy. The scikit- learn library provides an inbuilt feature for performing SVD which is np.linalg.svd() in NumPy. Its ability to handle any matrix, efficiently reduce dimensions, and focus on dominant features makes it very efficient.

Now we will see one mathematical example for SVD:

Consider a 2×2 matrix B ,

B = $\begin{matrix} 4 & 3 \\ 0 & -5 \end{matrix}$

STEP 1: Find $B^T$B

$$B^T = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix}$$

$B^T$B = $\begin{bmatrix} 16 & 12 \\ 12 & 34 \end{bmatrix}$

Step 2: compute the eigen values

|$B^T$B - λI| = 0 implies $\begin{vmatrix} 16-\lambda\mathrm{I} & 12 \\ 12 & 34-\lambda\mathrm{I} \end{vmatrix}$ = 0

We get $\lambda_1 = 40$, $\lambda_2$ = 10

The singular values are the square roots of the eigenvalues:

$\sigma_1$ = 6.3246 approx.

$\sigma_2$ = 3.1623 approx.

Step 3: find the eigenvectors of $B^T$B:

For $\lambda_1$ = 40 ,

($B^T$B- $\lambda_1$ $I$)v = 0

We get , -24x+12y=0 ⇒y=2x

Let x = 1 then y =2

Normalised eigenvector = $\begin{matrix} 0.4472 \\ 0.8944 \end{matrix}$

Similarly for $\lambda_2$ = 10,

($B^T$B- $\lambda_2 I$)v = 0

We get , 6x+12y=0 ⇒x=−2y

Let y=1, then x=−2

Normalised eigenvector = $\begin{matrix} -0.8944 \\ 0.4472 \end{matrix}$

$V^T$ = $\begin{bmatrix} 0.4472 & 0.8944 \\ -0.8944 & 0.4472 \end{bmatrix}$

Step 4: now find the left singular vectors

$u_i = \frac{Bv}{\sigma i}$

We get, U = $\begin{bmatrix} 0.8944 & 0.4472 \\ -0.4472 & 0.8944 \end{bmatrix}$

Step 5: construct sigma

∑ = $\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$ = $\begin{bmatrix} 6.3246 & 0 \\ 0 & 3.1623 \end{bmatrix}$

Therefore, SVD of B is:

$$B = U\sum V^T$$

B =$\begin{bmatrix} 0.8944 & 0.4472 \\ -0.4472 & 0.8944 \end{bmatrix} \begin{bmatrix} 6.3246 & 0 \\ 0 & 3.1623 \end{bmatrix} \begin{bmatrix} 0.4472 & 0.8944 \\ -0.8944 & 0.4472 \end{bmatrix}$

We can see that any matrix can be decomposed into three matrices using SVD .

# CHAPTER 3

## 3.1 EXTENSIONS AND VARIATIONS OF PCA IN MACHINE LEARNING

For many years, principal component analysis has been an essential technique in data analysis and to reduce dimensions. Originally introduced by Karl Pearson in 1901 as a method for identifying patterns in data, PCA then was formalized by Harold Hotelling in the 1930s, expanding its applications beyond statistics to fields like image processing, finance, and machine learning. The idea behind PCA is same is to transform high dimensional data into lower dimensional data with much information loss. However, standard PCA has encountered a number of challenges as datasets have become more huge and complicated, which has prompted the creation of numerous extensions and adaptations.

In the early days, main focus of PCA was to analyse small datasets with a large number of linear correlations. But with the increase in datasets in this huge space difficulties such handling non-linearity, robustness to outliers, sparsity, and processing efficiency have become critical. Traditional PCA is less useful in many real-world situations since it makes the assumption that data is linearly separable as well as subject to noise and outliers. To get over these problems, researchers have created a number of PCA variations, each of which is meant to meet a certain restriction.

The following are examples of modern PCA adaptations: Sparse PCA, which improves interpretability by choosing a subset of important features; Robust PCA, which improves stability by reducing sensitivity to noise and outliers; Kernel PCA, which improves PCA to non-linear datasets by mapping data into a higher-dimensional space using kernel functions; and Incremental PCA, which makes PCA efficient when applied to large-scale datasets by processing data in batches. These modifications have improved PCA's utility and usability as a tool in areas including image recognition, biology, and finance.

We will now go into more detail about these PCA innovations, including how they work and how they overcome the limitations of traditional PCA, in the sections that follow. Also we will look at the maths intuition behind all of these variations of PCA:

## 3.2 KERNEL PCA

Traditional PCA assumed that data is well represented in linear subspace, which limits its effectiveness when working with complex datasets where patterns are not easily separated by straight lines. KPCA uses a kernel function to get around this restriction by converting the data into a higher-dimensional feature space, which makes linear PCA possible.[4] This technique is also used in SVM when dataset is non separable and we want to transform lower dimension to higher dimension.

Kernel PCA maps the data into new space using a function, denoted as ϕ. To avoid the computational cost, KPCA uses kernel tricks which allows us to compute the dot product of transformed points without directly performing the transformation, in this we can avoid the large computations.

There are various kinds of kernel functions such as polynomial kernel, sigmoid kernel etc.

$\mathrm{K}(x_1 x_2) = \phi(x_1)^T\ \phi(x_2)$ , this is called kernel function.

Popular kernel functions include:

- **Polynomial kernel** - $\mathrm{K}(x_1 x_2) = (x_1^T x_2\ +\ \mathrm{c})^q$

Example : let $x_1 = [1\ 2]$ , $x_2 = [2\ 3]$ and c = 1

$\mathrm{K}(x_1 x_2) = (8 + 1)^2 = 81$

- **Gaussian kernel** - $\mathrm{K}(x_1 x_2) = e^{\frac{-\|x_1 - x_2\|^2}{2\sigma^2}}$

- **Sigmoid kernel** - $\mathrm{K}(x_1 x_2) = \tanh\ (\mathrm{a} x_1^T x_2\ +\ \mathrm{c})$

Now we need to find the kernel matrix M, whose elements are nothing but the dot product of transformed data points. The choice of kernel function plays a crucial role in this process, as it determines how the data is mapped. The choice of kernel function plays a important role in this process, as it determines how the data is mapped. After this, traditional PCA steps are applied: computing eigenvalues and eigenvectors of the centered kernel matrix, selecting the principal components based on the largest eigenvalues, and projecting the data onto these components to obtain a lower-dimensional representation.

KPCA comes with several limitations:

1. Which kernel function fits best – KPCA does not gives us any information about the best fit kernel . there is no universal rule to find this
2. Complexity - For large datasets, this leads to high memory usage and computational cost, making KPCA difficult for big data applications.

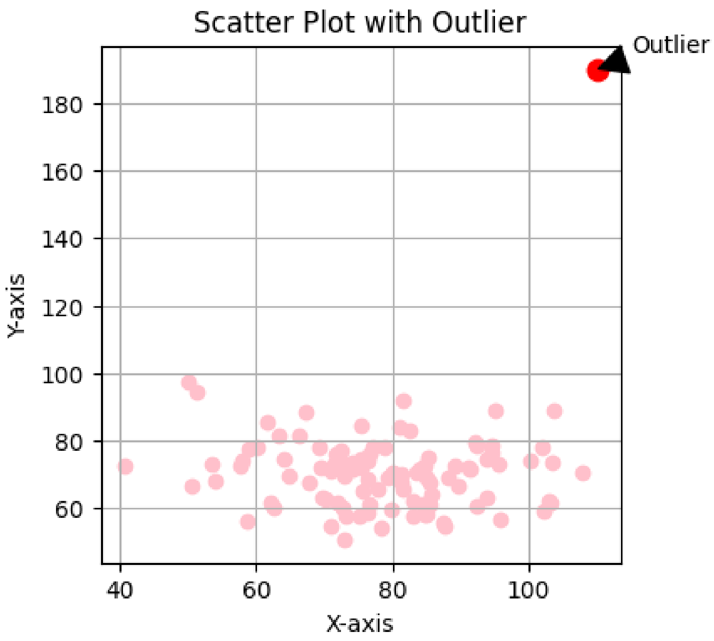
3. Problem in interpretation – In traditional PCA, where the principal components are linear combinations of original features, KPCA

   operates in a transformed high-dimensional space, making it difficult to interpret the results.
4. Lack of scalability - KPCA's performance drastically decreases for very big datasets because it depends on storing and computing the kernel matrix.

Applications of KPCA:

1. Use in biotechnology: KPCA is applied to genetically modify or detect patterns in genetic traits. In high-dimensional biological datasets, it improves DNA sequence clustering and classification, we can say it is used for gene data analysis.
2. To analyse whether and climate data: it helps us to analyse climate patterns by determining sea surface temperature.
3. Processing satellite images : it includes complex or detailed data and we get to know about the amount of aerosols which will help us to know about the cloud formation.
4. Recognition of images: It aids in identifying faces, recognizing handwritten text, and classifying objects.
5. Defining diseases and medical diagnoses: KPCA making it easier to detect diseases such as cancer and brain disorders.[10] It also helps in organizing and sorting patient records.

## 3.3 ROBUST PCA ( RPCA)



One of the key drawbacks of traditional PCA is its sensitivity to outliers and noise. Even a few corrupted data points can significantly affect the principal components, leading to inaccurate feature extraction and poor performance in real-world applications. Robust PCA (RPCA) was introduced to address this limitation by ensuring that the extracted components remain stable and reliable, even when dealing with noisy or misclassified data.[11]

The core idea behind RPCA is to decompose a given data matrix M into two components:

$$\mathbf{M = L + R}$$

L is a Low-Rank Matrix: Represents the clean and structured part of the data, capturing the underlying patterns while removing out noise.

R is a Sparse Matrix: Contains the corruptions, outliers, or noise, ensuring that these anomalies do not distort the meaningful features.

A low-rank matrix is essential because real-world data often lies in a lower dimensional subspace despite being represented in a high-dimensional form. For example: Facial photos of the same person taken in various lighting situations still have similar structures, which means that a low-rank representation can capture their variation in face recognition. By focusing on the low-rank structure, RPCA preserves the most informative part of the data while discarding noise and outliers. Now, we can think that there infinite number of ways we can decompose a matrix into sum of other two matrices then how can this method be efficient or how can we select the most appropriate decomposition. so the answer is that we can done this by putting some constraints and converting this into a convex optimization problem. let us discuss that now:

To effectively separate the low-rank and sparse components, RPCA formulates the decomposition as an **optimization problem** with specific constraints. The goal is to minimize the rank of L and the sparsity of R while ensuring that their sum reconstructs the original matrix M. Since directly minimizing the rank function is computationally challenging, an alternative approach is used:

$$\min \| L \|_{*} + \lambda \| R \|_{1} \quad \text{subject to} \quad M = L + R$$

where,

$\| L \|_{*}$ = The sum of singular values of L, used as a relaxation of rank minimization (nuclear norm).

$\| R \|_{1}$ = Encourages sparsity in R, ensuring that only a few entries contain significant noise or outliers (L1 norm).

$\lambda$ = A weighting parameter that balances the trade-off between keeping data clean and preserving key structures.

But RPCA has various limitations, such as it takes lot of computational cost as solving the optimization problem requires repeated SVD, making it inefficient for large-scale datasets. Another problem is that this problem is highly dependent on choice of λ, wrong choice may lead to poor results, this plays a crucial role in RPCA. Also RPCA considers that there is always a noise in the data, but it is not always in the case of highly correlated data. These limitations highlight the need for alternative PCA variations, such as Sparse PCA, Kernel PCA, and Incremental PCA, to address specific challenges more effectively.

## 3.4 SPARSE PCA (SPCA)

Question arises is why we need sparse PCA when we already have other variants of PCA?

As clear from the name "sparse" means few or scattered. In PCA, assume we have large dataset for example let's say we have dataset with 200 features and according to traditional PCA it creates new principal components that are linear combinations of all 200 features. So when there is lack of sparsity, researchers gives us the idea of sparse PCA.

To be precise, lack of sparsity makes interpretation and feature selection very difficult. Instead of finding the most important features, traditional PCA will start to work on all features

Now sparse PCA will help to reduce noise and redundant that is those features which are not important. Now I think we get the idea why we need sparse PCA, along with that we are going to understand the maths intuition behind sparse PCA in the following section:

$$\min \frac{1}{2}||A - AX||_F^2 + \alpha||X||_1$$

where, A be the data matrix and X be the principal component matrix. Let us breakdown the formula -

The first term ensures that reconstructed data AX would be close to original data, reducing the error. And in the second term L1 norm is used as it makes models sparse by setting some values to zero. That means only few components contribute to principal components. The constant α controls the level of sparsity, big α means more sparsity and small α means less sparsity. After that $||.||_F^2$ means the Frobenius norm, which measures the squared error between original data and reconstructed data. This is all about the formula of sparse PCA published in one of the paper in 2006 .

Now, we will see how to apply Sparse Principal Component Analysis (Sparse PCA) using Python's scikit-learn library.

There are various drawbacks of sparse PCA:

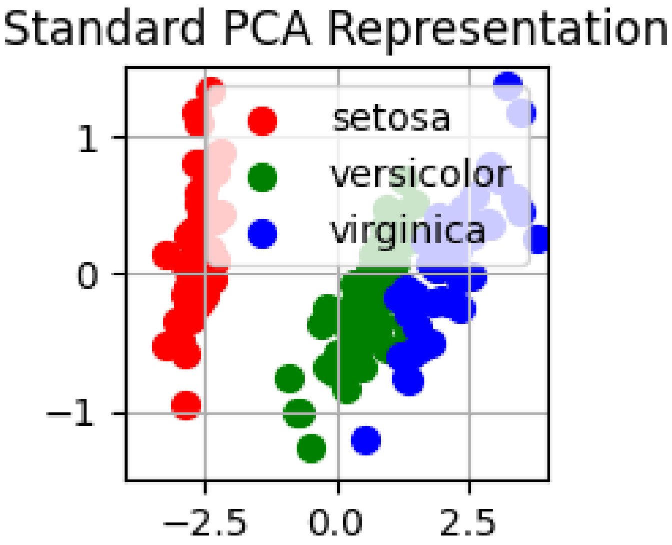
1. Role of α is crucial – The performance is highly influenced by the choice of α. Degree of sparsity is determined by this, either too much sparsity result in losing some important features or very less sparsity.
2. Lack of information about variance - It may remove significant variance from the original data, which could result in less than ideal feature selection.
3. Complexity – As compared to traditional PCA, sparse PCA is costly due to optimization involved.
4. Not ideal for selecting features - Sparse PCA might not be the best option always for selecting the features because it forces sparsity in its main parts rather than specifically choosing features.

## 3.5 INCREMENTAL PCA

When working with huge dataset, storing and managing those datasets is not really easy task time all the time. To solve this problem, Incremental PCA is introduced by researchers. It is a better version of traditional PCA made to solve the problem of managing large scale data. The process is IPCA transform into small groups unlike traditional PCA.

As new data becomes available, IPCA iteratively updates the principal components rather than computing them all at once. This method maintains the capacity to identify significant patterns in high-dimensional data while reducing memory needs.

By continuously refining the principal components, IPCA ensures that it can adapt to evolving datasets without needing to recompute everything from scratch. This makes it highly effective for real-time applications, online learning, and large-scale data analysis.



This code shows the comparision between the standard PCA and incremental PCA. Both approaches are used to split the dataset into two principal components, and the outcomes are shown for comparison. IPCA is best for large datasets , as it breaks the dataset into small batches . PCA and IPCA produce comparable outcomes because of the small Iris dataset.

# CHAPTER 4

## 4.1 PCA in Bioinformatics and Genome Analysis

Use case 1 – PCA FOR PLANT STRESS :

Any external factor that interefers with a plant's regular growth , development, or physiological functions is referred to as plant stress. There are two types of plant stress , biotic stress - which is caused by living organisms ( bacteria , Insects and pests, Weeds and competing plants) , abiotics stress - Caused by non-living environmental factors (draught , salinity , UV radiations) .[6]

We will understand how PCA contributes in managing plant stress :

Principal Component Analysis plays a crucial role in analyzing large datasets by identifying underlying patterns and reducing dimensionality. By eliminating irrelevant or redundant features, PCA allows scientists to focus only on the most significant factors, improving the efficiency of research and decision-making. One key application of PCA in agriculture is assessing salt tolerance among plants. PCA is widely used in studying gene expression patterns which allows researchers to determine which particular genes are beneficial in reducing environmental stress. Furthermore, PCA helps in remote sensing applications, enabling early detection of plant stress before visible damage occurs, this early approach allows for immediate action by improving crop management and ensuring sustainable agricultural practices. By analyzing plant traits and gene expressions, PCA helps identify genetic variations associated with stress resistance which enables the selection of strong plant varieties which in turn contributes to reducing plant stress and indirectly enhances soil fertility by encouraging healthier crop growth.

Scientists use biotechnology to enhance plant resistance to stress through very useful CRISPR-Cas Technology –

By enabling researchers to precisely edit genes, examine gene functions, and categorize various bacterial strains, CRISPR-Cas technology has altered the study of bacterial genomes. It aids in the identification and modification of particular genes, improving our knowledge of bacterial evolution, pathogenic potential, and antibiotic resistance. However, because of the vast number of variables and complex patterns, CRISPR-based investigations produce a lot of complex genomic data that is challenging to analyze. That is where PCA commes in role , it helps in interpret and process large scale bacterial genomic data . PCA reduces the number of features while maintaining essential genetic patterns. PCA helps researchers to focus on the key features which would help them to study more about genetic patterns.[12] It will help to classify same kind of genes by making clusters . It helps in visually representing bacterial genome variations, making it easier to identify groups of similar bacteria or unusual strains with unique genetic changes.

Use case 2 – PCA FOR FINDING PROTEIN STRUCTURE:

Protein structure means specific shapes based on the arrangements of amino acids. To investigate protein structures, learn about molecular dynamics, and forecast folding patterns, scientists use computer methods such as PCA. It majorly helps in to study about the flexibility and movements of proteins . let us know more about how PCA in helpful in finding protein structure –

1. PCA helps to reduce the dimesions as there are thousands of atomic coordinates , because of this researchers can efficiently work on the most significant structural changes.
2. It is used to cluster similar types of genes, which helps in the classification of genetic patterns. Researchers can identify various genetic differences, find possible biomarkers, and differentiate between healthy individuals because to this clustering.
3. PCA helps scientists study how proteins change shape when a molecule binds to them. By analyzing these movements, researchers can identify important regions where drugs can attach.

There is one study called Alzheimer disease gene study : A progressive neurological condition that affects thinking, behavior, and memory is Alzheimer's disease. It results from the accumulation of unusual protein structures in the brain, like tau tangles and beta-amyloid plaques, which harm nerve cells and impair their ability to communicate with one another. Now PCA when combined with other machine learning algorithm can help to cure this disease or hepful in better diagnosis with its properties such as by reducing the dimensions as it has thousands of genes .[7]

## 4.2 PCA IN COMPUTER VISION AND IMAGE PROCESSING

PCA can solve various challenges in computer vision such as face recoginition, handwriting recognition, image compression etc. Various PCA variations improve its performance in computer vision by tackling certain issues such as sparsity, robustness, and non-linearity. For example KPCA, it is a useful tool for identifying difficult, nonlinear patterns in photos and extensively utilized in picture segmentation, texture classification, and object recognition.

USE CASE 1 : PCA for Autonomous Surveillance and Threat Detection in Defense

In modern defense technology, real-time thermal imaging is critical for detecting unauthorized movements, potential threats, and enemy activities in low-visibility conditions. PCA speeds up processing by lowering the dimensionality of the data, which makes it possible to identify defects more quickly. To effectively identify and monitor threats, a PCA-based system may be installed on ground-based infrared cameras or surveillance planes.[8] It also helps to eliminate background noise, ensuring that only relevant thermal signatures are examined for increased threat detection accuracy.

1. Since infrared rays include huge quantities of data, PCA eliminates duplicate features, improving quick processing and real-time decision-making.

2. PCA reduces false alarms by taking important thermal features to differentiate between people, animals, and vehicles; it also improves recognition of targets by concentrating on the most useful thermal patterns, which increases reliability and precision of surveillance by eliminating redundant information.

3. It compresses the thermal video so that video can stream smoothly over all places like in every military area or even in remote area where connectivity is less.

4. To react quickly to any threats, this can initiate security steps or send notifications automatically, PCA helps to analyse thermal data to find different-different patterns.

PCA is a powerful defensive technology tool because it improves awareness of conditions, surveillance accuracy, and real-time danger detection when combined with thermal imaging and artificial intelligence.

## USE CASE 2 : PCA on Contour Detection using OpenCV

In computer vision, contour detection method is used to determine the boundaries of any object. Using PCA, we are able to find the looks of any object such as shape, alignment etc. PCA helps in identifying the contour's major and minor axes which is helpful for picture preprocessing, object alignment, and shape recognition.

OpenCV means open source computer vision library, in this library there are specific types of tools which are helpful in finding the contour.

When it comes to contour detection in OpenCV, PCA assists in determining the current contour direction in an image, which is helpful for alignment, object recognition, and shape analysis.

# CHAPTER 5

## Challenges, Limitations and Future Scope

While PCA has various advantages, it is not always the perfect solution for every problem. Researchers often face several challenges when applying PCA in experiments. One major limitation is its assumption of linearity, meaning it may not perform well when dealing with complex, nonlinear relationships in data. Additionally, PCA is highly sensitive to noisy or irrelevant features, which can lead to misleading results. Another challenge is the difficulty in interpretability of new components, as they are linear combinations of the original variables rather than direct meaningful features. However, choosing the optimal number of principal components is not always straightforward, and selecting too many or too few can impact the effectiveness of dimensionality reduction. We will study these limitations one by one in detail:

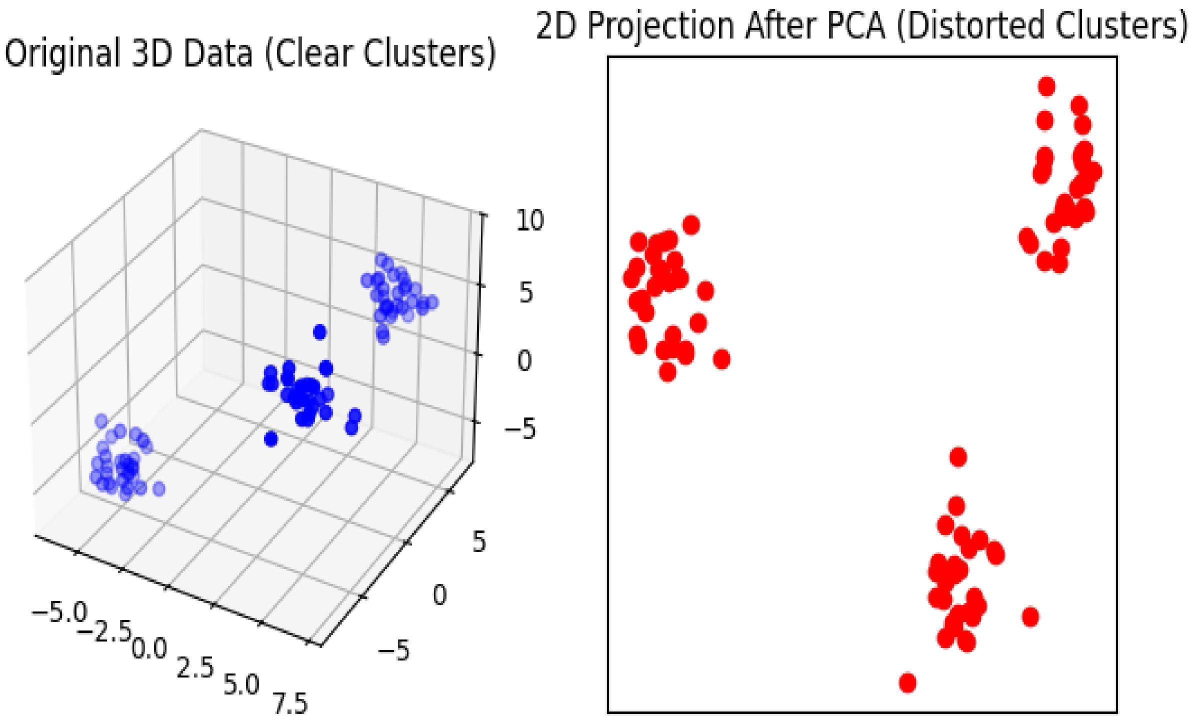
## 5.1 CHALLENGES IN HANDLING HIGH DIMENSIONAL AND NOISY DATA

When using PCA, managing complex and noisy data presents major difficulties. Finding the principal components that are actually relevant in high-dimensional datasets can be challenging due to the quantity of related features. Even though PCA helps in dimension reduction, it keeps elements that don't add much to the total variance, which results in poor feature selection. Furthermore, because PCA cannot naturally differentiate between important variations and random fluctuations, noise in the data can have a major impact on its performance. The performance of the extracted components may be decreased if PCA detects and amplifies noisy elements. Another challenge is the "curse of dimensionality," where an increase in the number of features leads to computational inefficiencies and requires more data to maintain results. Moreover, interpreting the transformed components becomes difficult as they are linear combinations of multiple original features rather than straightforward variables. To address these issues, researchers often use preprocessing techniques such as feature extraction, normalization, and methods to reduce noise before applying PCA to ensure the extracted principal components are both relevant and interpretable.

The example of thermal photography in defense technologies helps us better understand. Assume that a surveillance drone monitors a border region using infrared cameras to spot any unauthorized activity. A high-dimensional dataset is produced by the hundreds of pixels with different patterns of heat found in the thermal photos that were taken. In order to identify possible dangers, such as differentiating between people, animals, or cars, PCA is used to reduce dimensionality and extract essential data.

However, the accuracy of PCA may be affected by data noise, which can be brought on by outside variables like fog, rain, or sensor failures. PCA may incorrectly consider noise as a significant feature since it lacks the inherent ability to distinguish between relevant signal changes and random noise. For example, PCA might consider abrupt temperature changes brought on by wind or heat reflections from surfaces as principal components, which could result in false positives. This may lead to wrong warnings or an inability to identify real dangers.

Additionally, PCA may remain computationally costly and reduce real-time efficiency in high-dimensional thermal video streams if it retains an excessive number of components. However, important thermal patterns that help in threat identification may be missed if insufficient components are chosen. Therefore, even though PCA aids in feature extraction and data compression, appropriate preprocessing—such as removing noise and choosing pertinent components—is required to guarantee its efficacy in these practical applications.



The original data with distinct groups is displayed in the plot on the left. The right plot shows how the clusters may overlap or become distorted after using PCA to decrease the dimensionality to 2D, losing important structure in the process. This illustrates how, when lowering dimensions, PCA can occasionally distort significant patterns.

## 5.2 Limitations of PCA in nonlinear transformations.

PCA functions best when the significant variations in the dataset show straight-line relationships, which is based on the assumption that the data is linearly separable. However, PCA is unable to represent the complex, nonlinear structures found in many real-world datasets. PCA may lose important patterns when applied to such data, which could result in the loss of important information. For instance, curved or nonlinear manifolds are frequently followed by object shapes or facial expressions in picture recognition. PCA may misrepresent these variations by using only linear projections, which would lower the classification accuracy.

Let us understand this by an example , Remote sensing images such as satellite imagery, often contain highly nonlinear patterns due to variations in terrain, atmospheric conditions, and object shapes. PCA, being a linear technique, struggles to capture these complex relationships. Consider this scenario where we have to find land cover types from satellite images such as distinguishing between forests, water bodies, urban areas, and agricultural land. The way different land types reflect light creates complex patterns in high-dimensional space. These patterns are influenced by factors like how light hits the surface, changes in brightness, and differences between seasons.

Challenges with PCA –

1. PCA tries to project this complex dataset onto a lower-dimensional space using straight-line projections, assuming that the most important variations are linear.

2. This often results in different land cover types overlapping in the reduced space which makes classification less accurate.

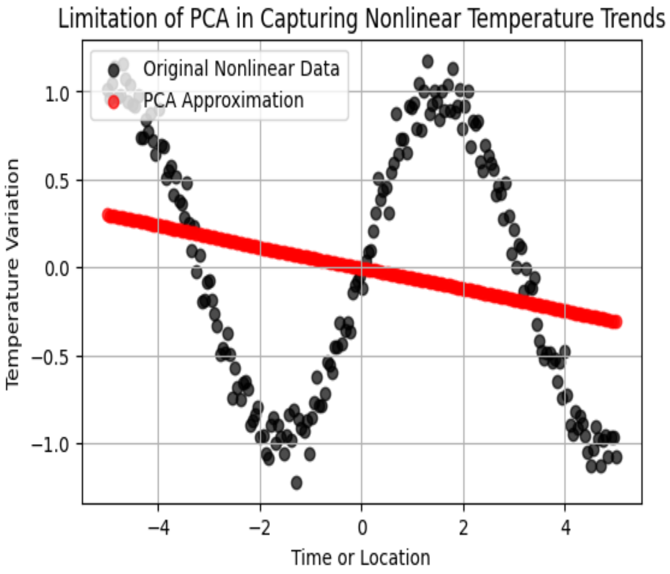
3. Nonlinear relationships, such as vegetation indices reacting differently to soil moisture levels may be lost after PCA compression.

Due to this there are various consequences such as different types of land cover (such as water, vegetation, and urban areas) may become harder to classify. Since PCA reduces the number of features, some important differences between these land types might get mixed together causing errors in classification. Essential features that help in distinguishing different surface types (e.g., subtle water pollution variations or shadow effects in urban areas) may not be well preserved.

Another great example explaining the limitation of PCA in non- linear transformation is Climate Statistics and Temperature Changes.

PCA assumes linearity, but many real-world datasets (such as climate data) exhibit nonlinear relationships. Temperature, humidity, and atmospheric pressure do not follow a simple linear pattern. PCA assumes that the most important variations in climate data are along straight-line directions, which is often not the case.

When scientists analyse global temperature datasets, they observe long-term warming trends mixed with short-term fluctuations caused by ocean currents, volcanic eruptions, or solar cycles. PCA can show an overall rise in temperature, but it misses important details, like the fact that polar regions heat up faster than tropical areas.



This image shows how PCA has difficulty capturing nonlinear temperature changes in climate data. The original temperature trend (black points) naturally follows a curved shape, representing complex variations over time. However, PCA (red points) tries to fit this data into a straight-line pattern, simplifying the trend and losing important details. This demonstrates one of the main limitations of PCA—it struggles to accurately represent nonlinear transformations, which are common in climate statistics.

## 5.3 Possible Advancements In Hybrid Approaches And Deep Learning.

As we know that Principal component analysis is the technique majorly used for dimensionality reduction. But in this section, we will try to understand PCA with other high dimensional techniques. In recent years, the combinations with PCA has proved or solved many problems. Furthermore, these advancements help in reducing dimension obviously, reducing noise and better feature extraction since data in this huge space is very complex. Now what could be the possible combinations, for example PCA combined with cluster analysis, neural networks, linear discriminant analysis etc. If we look deeper and thought more about these combinations, we can say that PCA could possibly combine with any other technique which we used in machine learning for improved accuracy. We will study more about these advancements in following section:

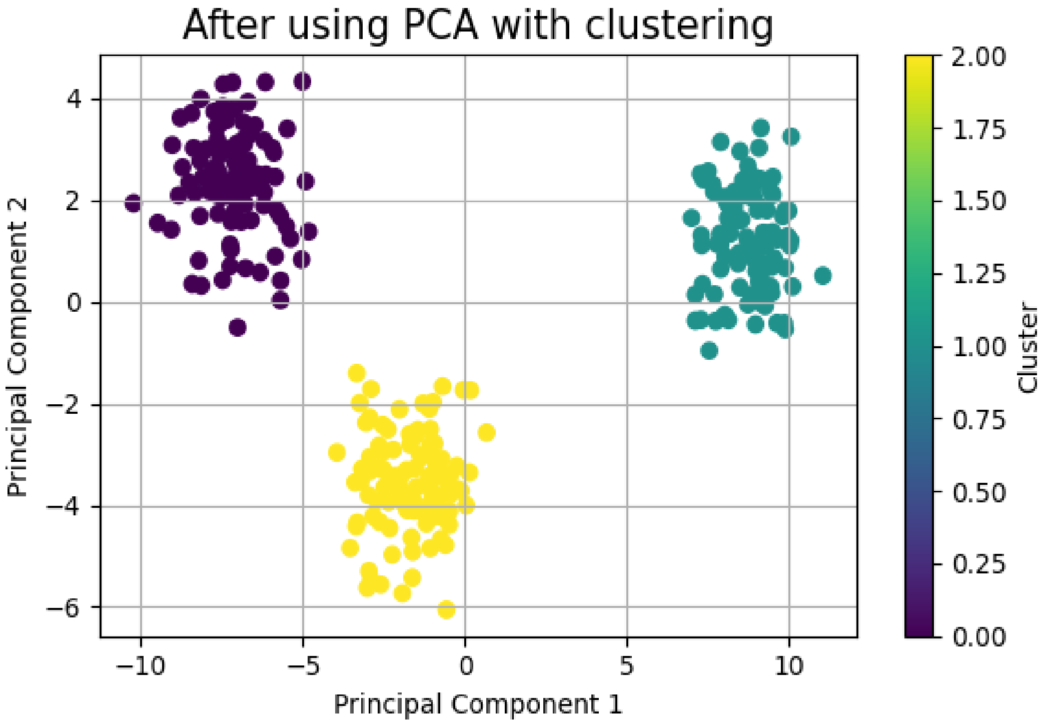
### 5.3.1 PCA Combined With Clustering

In cluster analysis, we try to put similar kind of items together according to their behaviour. We don't know the group in advance, unlike linear discriminant analysis where we already know the groups and want to know what really differentiates them. The main goal is to allow the machine to put together the similar kind of data that makes sense. Now clustering is used after PCA for dimensionality reduction.[3]

For example: In a carton, there are bulbs and tubes. Clustering helps to sort them into similar kind of things in one cluster and other kind in different group just by looking at their characteristics. And what PCA does is , since there are lot of dimensions like shape, colour, guarantee time etc. so PCA takes all these features and try to reduce them.

We will now understand more about this, PCA converts the data into two dimensional, in this way we can easily visualize the similarities by doing grouping in 2-d. but sometimes what happens is that reduced plot may miss some important group patterns. But using clustering this is not a bigger deal as we are trying to find patterns and don't already know them. PCA will give useful view if data doesnot have clear clusters.

EXAMPLE: Consider we have high dimensional dataset containing different features from different people such as height, weight, age, address etc. now the data looks unrelated and scattered. So first PCA is applied to reduce the feature and keeping only the useful features. After this we use clustering that will do the grouping of similar kind of features.



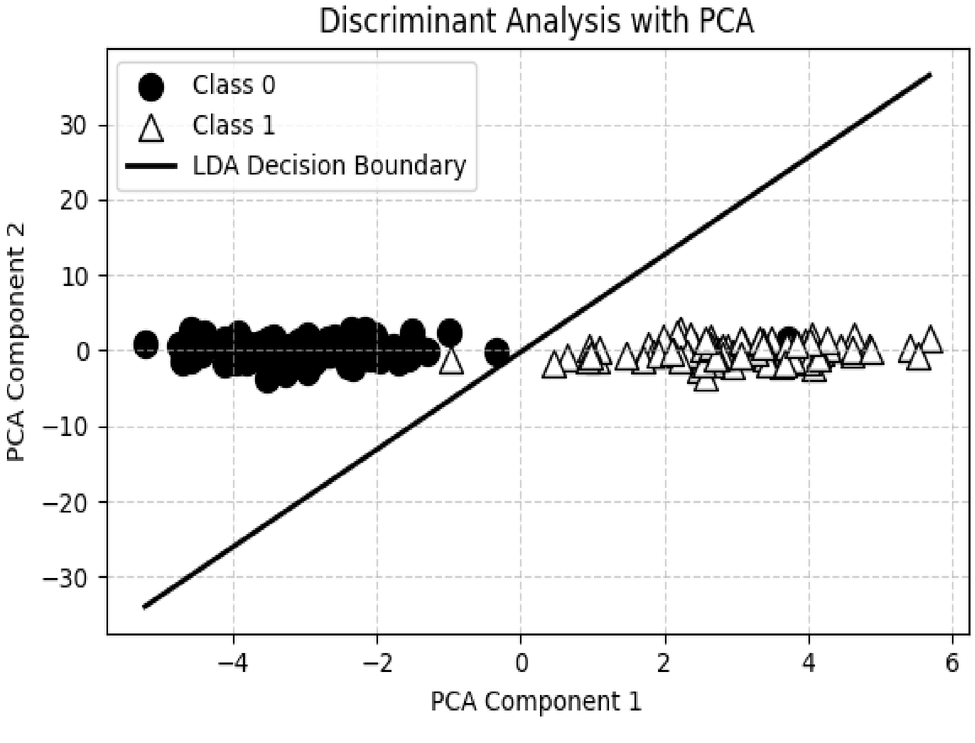
In the above graph, we can see that this is the more simplified and understandable 2-d plot. Even though the data set does not have any patterns, PCA helped to find the clusters. Hence, this technique helps to reduce dimensions and uncover the hidden groups, by which we can understand the visualization. Such types to methods basically used in image recognition, bioinformatics etc.

### 5.3.2 PCA Combined with Discriminant Analysis

Linear discriminant analysis is basically used for reducing dimensions. To separate the classes within the dataset by identifying linear combination of features. So the random vector r(say), assumed to have normal distribution. In linear discriminant analysis, our main goal is to minimize the problematic areas such as mis-classification.

Now since there are huge datasets, therefore there are lot of random vectors associated with it. Our idea of combining PCA with discriminant analysis is to replace these lot of random vectors by the first k(say) principal components whose variance are high.

Which eventually will led to reduction in dimensions, more accurate results and reduction of outliers.
We can clearly observe from the picture, that there are two separable classes by the decision boundary of discriminant analysis and then after applying PCA, there are two easily distinguishable clusters. Hence, with this hybrid approach we can reduce various problems faced by the researchers.



## References

[1] Buntine, Wray. "Variational extensions to EM and multinomial PCA." *European Conference on Machine Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[2] Pattern Recognition and Machine Learning by Christopher M. Bishop.

[3] I.T Jolliffe Principal Component Analysis, second edition.

[4] Hoffmann, Heiko. "Kernel PCA for novelty detection." *Pattern recognition* 40.3 (2007): 863-874.

[5] Ebied, Hala M. "Feature extraction using PCA and Kernel-PCA for face recognition." *2012 8th International Conference on Informatics and Systems (INFOS)*. IEEE, 2012.

[6] Sivakumar, J., et al. "Principal component analysis approach for comprehensive screening of salt stress-tolerant tomato germplasm at the seedling stage." *Journal of biosciences* 45 (2020): 1-11.

[7] López, Míriam, et al. "Principal component analysis-based techniques and supervised classification schemes for the early detection of Alzheimer's disease." *Neurocomputing* 74.8 (2011): 1260-1271.

[8] Arivudainambi, D., Varun Kumar KA, and P. Visu. "Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance." *Computer Communications* 147 (2019): 50-57.

[9] Weenink, David. "Canonical correlation analysis." *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam*. Vol. 25. Amsterdam: University of Amsterdam, 2003.

[10] Pechenizkiy, Mykola, Alexey Tsymbal, and Seppo Puuronen. "PCA-based feature transformation for classification: issues in medical diagnostics." *Proceedings. 17th IEEE Symposium on Computer-Based Medical Systems*. IEEE, 2004.

[11] Xu, Huan, Constantine Caramanis, and Sujay Sanghavi. "Robust PCA via outlier pursuit." *Advances in neural information processing systems* 23 (2010).

[12] Feng, Xuping, et al. "Discrimination of CRISPR/Cas9-induced mutants of rice seeds using near-infrared hyperspectral imaging." *Scientific reports* 7.1 (2017): 15934.

[13] Duda, Richard O., Peter E. Hart, and David G. Stork. "Pattern classification. 2nd edn Wiley." *New York* 153 (2000).

[14] Ye, Jieping, and Bin Yu. "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems." *Journal of Machine Learning Research* 6.4 (2005).

[15] Xu, Rui, and Donald Wunsch. "Survey of clustering algorithms." *IEEE Transactions on neural networks* 16.3 (2005): 645-678.

[16] Alpaydin, Ethem. *Introduction to machine learning*. MIT press, 2020

[17] Ross, David A., et al. "Incremental learning for robust visual tracking." *International journal of computer vision* 77 (2008): 125-141.

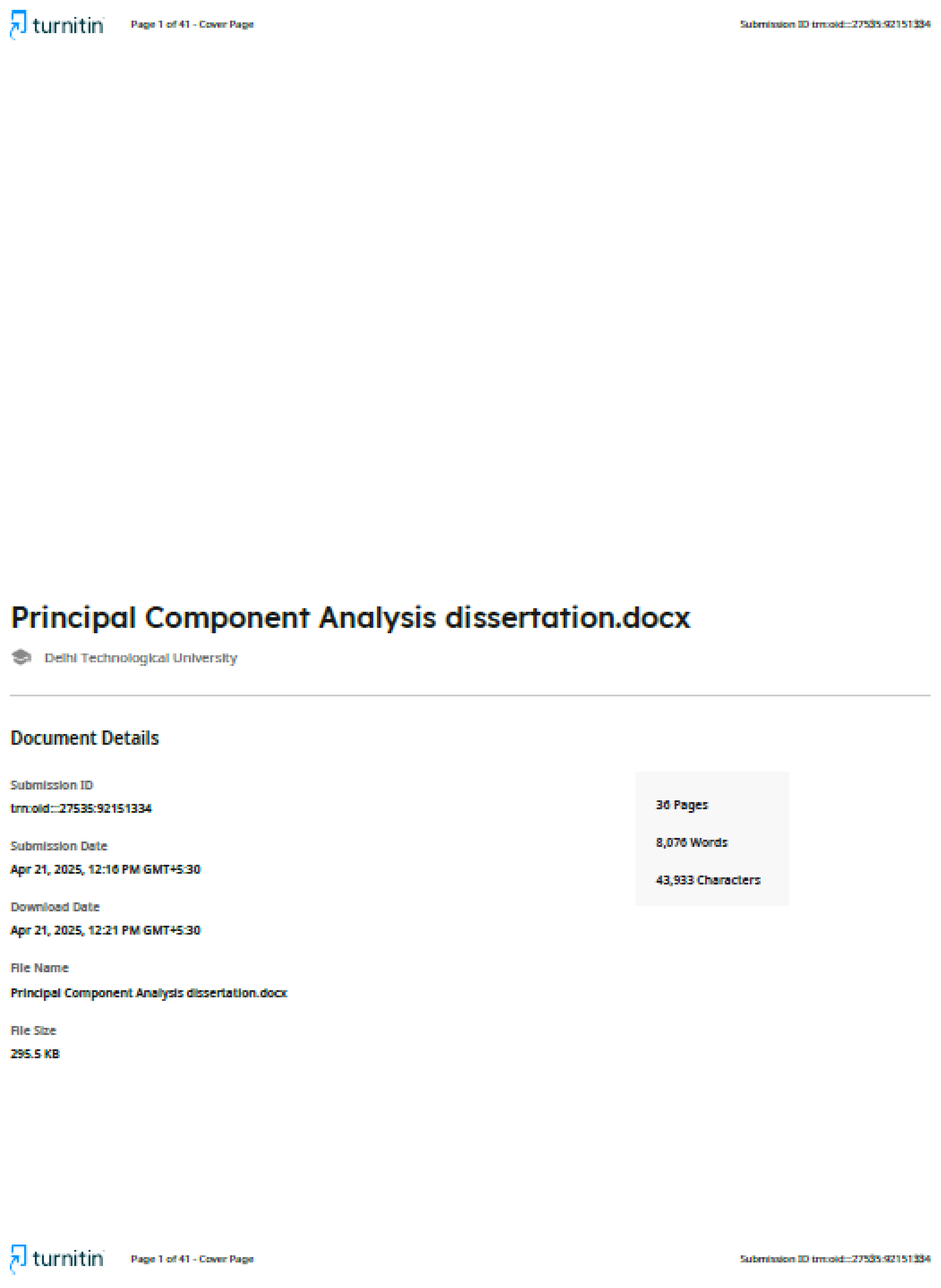
[18] Schölkopf, Bernhard, et al. "Kernel PCA pattern reconstruction via approximate pre-images." *International Conference on Artificial Neural Networks*. London: Springer London, 1998.

[19] Shawe-Taylor, John, and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[20] Abdi, Hervé, and Lynne J. Williams. "Principal component analysis." *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010): 433-459.

[21] Golub, Gene H., and Charles F. Van Loan. *Matrix computations*. JHU press, 2013.

[22] Eckart, Carl, and Gale Young. "The approximation of one matrix by another of lower rank." *Psychometrika* 1.3 (1936): 211-218.



# Principal Component Analysis dissertation.docx

Delhi Technological University

## Document Details

Submission ID

trn:oid:::27535:92151334

Submission Date

Apr 21, 2025, 12:16 PM GMT+5:30

Download Date

Apr 21, 2025, 12:21 PM GMT+5:30

File Name

Principal Component Analysis dissertation.docx

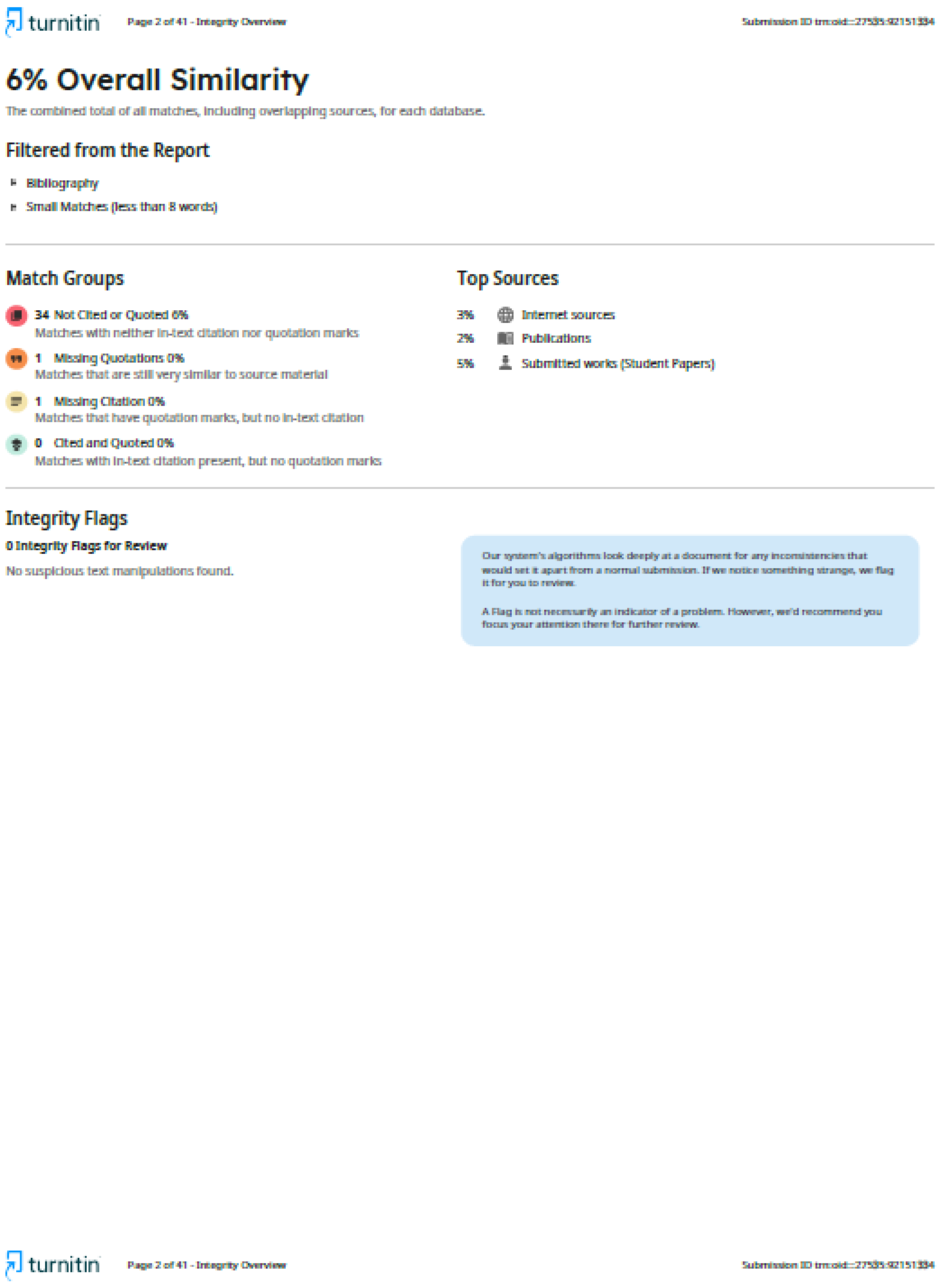
File Size

295.5 KB

36 Pages

8,076 Words

43,933 Characters



turnitin Page 2 of 41 - Integrity Overview Submission ID trn:oid:::27535:92151334

# 6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Filtered from the Report

- Bibliography
- Small Matches (less than 8 words)

## Match Groups

- **34** Not Cited or Quoted 6%
  Matches with neither in-text citation nor quotation marks
- **1** Missing Quotations 0%
  Matches that are still very similar to source material
- **1** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

## Top Sources

| | |
|---|---|
| 3% | Internet sources |
| 2% | Publications |
| 5% | Submitted works (Student Papers) |

## Integrity Flags

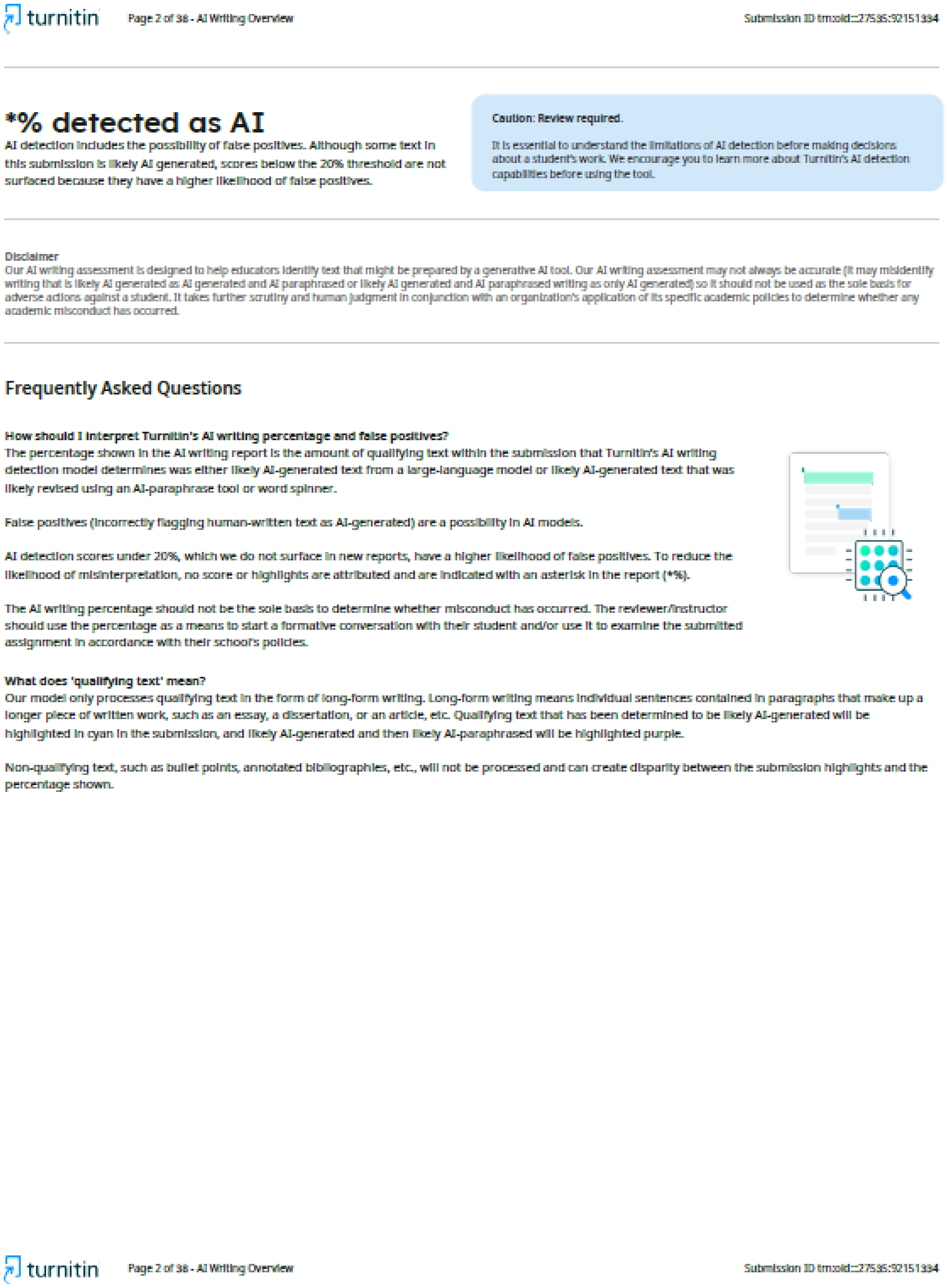
**0 Integrity Flags for Review**

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 41 - Integrity Overview Submission ID trn:oid:::27535:92151334



turnitin Page 2 of 38 - AI Writing Overview Submission ID trn:oid:::27535:92151334

## *% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

**Caution: Review required.**

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

**Disclaimer**
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

### Frequently Asked Questions

**How should I interpret Turnitin's AI writing percentage and false positives?**
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

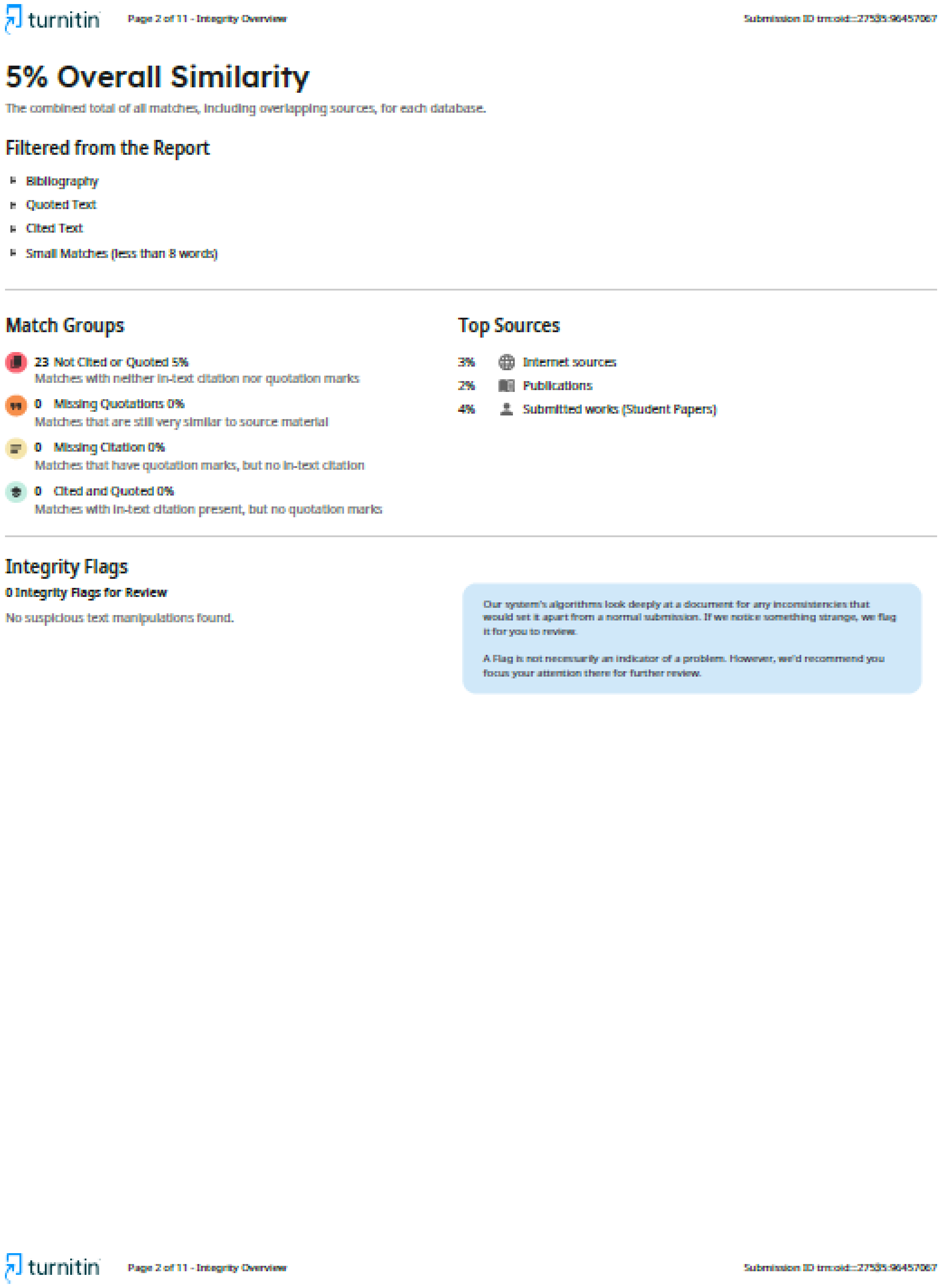
AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

**What does 'qualifying text' mean?**
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

turnitin Page 2 of 38 - AI Writing Overview Submission ID trn:oid:::27535:92151334



turnitin Page 2 of 11 - Integrity Overview Submission ID trn:oid:::27535:96457067

# 5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

## Match Groups

23 Not Cited or Quoted 5%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

## Top Sources

| | |
|---|---|
| 3% | Internet sources |
| 2% | Publications |
| 4% | Submitted works (Student Papers) |

## Integrity Flags

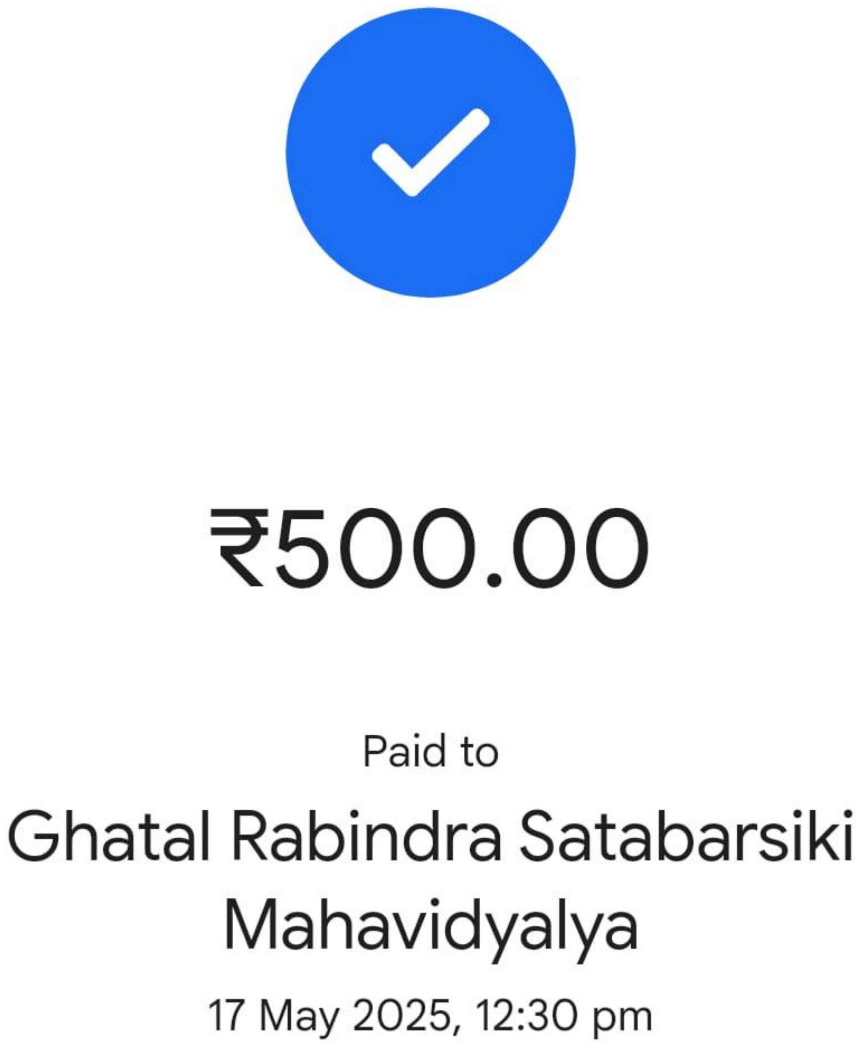
**0 Integrity Flags for Review**

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 11 - Integrity Overview Submission ID trn:oid:::27535:96457067



₹500.00

Paid to

Ghatal Rabindra Satabarsiki Mahavidyalya

17 May 2025, 12:30 pm