Lab Report: Matrix Decomposition

Name: CAO Xinyang
ID: 321793

1. Introduction

This lab explores the application of matrix decomposition techniques in machine learning. The primary focus is on understanding how Principal Component Analysis (PCA), Fast Independent Component Analysis (FastICA), and Non-Negative Matrix Factorization (NMF) can be used for dimensionality reduction and their impact on model performance.

2. Objective

1. Learn and implement basic matrix decomposition techniques: PCA, FastICA, and NMF.
2. Analyze their performance on the Optical Recognition of Handwritten Digits dataset.
3. Evaluate the performance of a Random Forest classifier on original and reduced data.
4. Derive insights into the benefits and challenges of dimensionality reduction for machine learning models.

3. Dataset Overview

Dataset: Optical Recognition of Handwritten Digits (UCI Repository).
Description: The dataset consists of 64 features (pixel intensities of 8x8 images) and 10 target classes (digits 0-9).
Train/Test Split: Predefined splits from optdigits.tra and optdigits.tes files.

4. Methodology

4.1 Preprocessing

- Features were standardized using StandardScaler for PCA and FastICA.
- For NMF, features were normalized to the [0, 1] range using MinMaxScaler.

4.2 Dimensionality Reduction

- Applied PCA, FastICA, and NMF to reduce features from 64 dimensions to:
  - **10 components** (default for consistent comparison).
  - **30 components** (extended PCA to retain more variance).
- Added Linear Discriminant Analysis (LDA), a supervised method, reducing to 9 dimensions (target class count - 1).

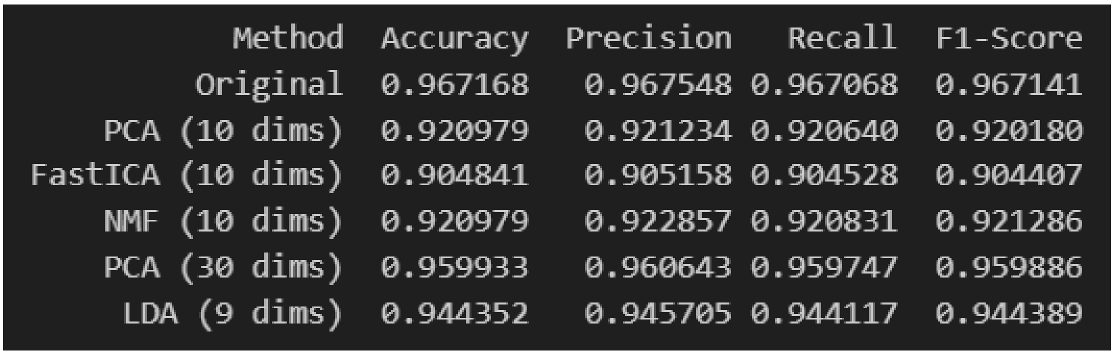
4.3 Classification

- Random Forest was used as the classifier with default parameters.

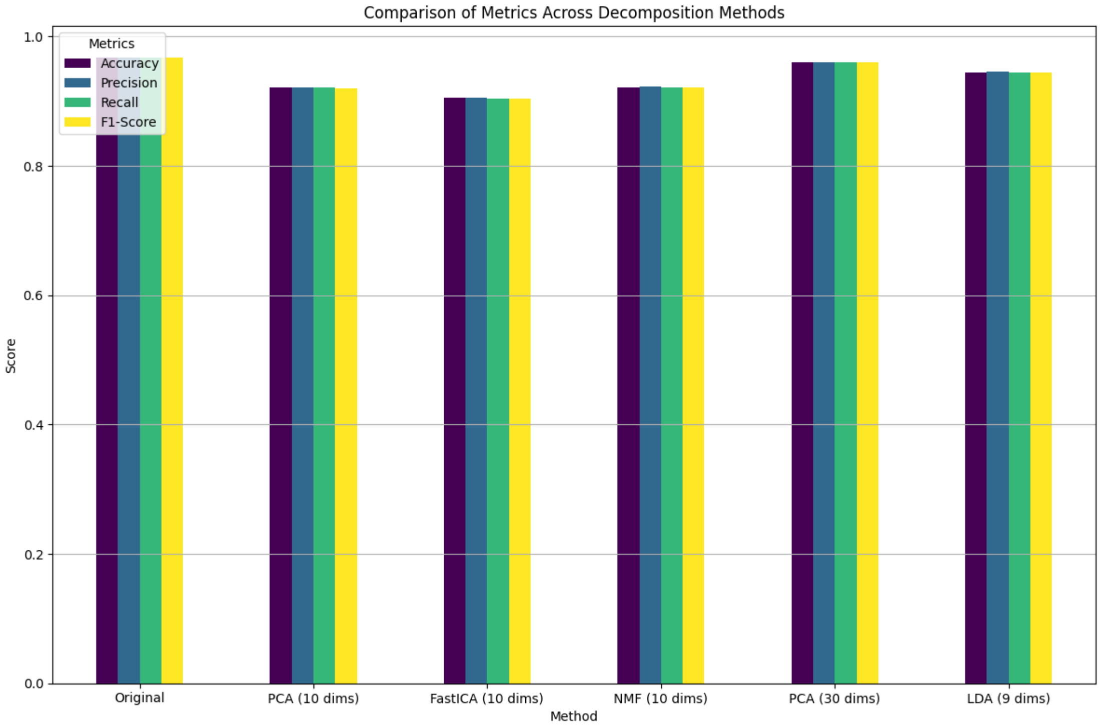
- Metrics evaluated: **Accuracy**, **Precision**, **Recall**, and **F1-Score**.

5. Results

The performance of the Random Forest classifier on the original and reduced datasets is summarized below:



```
           Method  Accuracy  Precision   Recall  F1-Score
         Original  0.967168   0.967548 0.967068  0.967141
    PCA (10 dims)  0.920979   0.921234 0.920640  0.920180
FastICA (10 dims)  0.904841   0.905158 0.904528  0.904407
    NMF (10 dims)  0.920979   0.922857 0.920831  0.921286
    PCA (30 dims)  0.959933   0.960643 0.959747  0.959886
     LDA (9 dims)  0.944352   0.945705 0.944117  0.944389
```



6. Analysis

6.1 Insights into Matrix Decomposition Techniques

- **PCA**:
  - PCA with 10 components retained approximately 92% accuracy, demonstrating its ability to retain variance effectively.
  - Increasing PCA dimensions to 30 significantly improved accuracy to 96%, closely matching the original dataset's performance.
  - PCA is suitable for datasets with high redundancy and variance-driven features.
- **FastICA**:

    - FastICA resulted in slightly lower accuracy (90%) compared to PCA and NMF.
    - It focuses on maximizing statistical independence, which may not align with variance or class separability for classification tasks.
- **NMF**:
    - NMF performed comparably to PCA (10 components) with 92% accuracy.
    - It is particularly useful for non-negative data but may require tuning for optimal results.
- **LDA**:
    - LDA achieved 94% accuracy with only 9 dimensions, leveraging supervised learning to preserve class separability.
    - It is particularly effective for classification but less flexible than unsupervised methods.

6.2 Observations

- **Dimensionality Reduction vs. Model Performance**:
    - Reducing dimensions from 64 to 10 led to some performance degradation across all methods.
    - Retaining more dimensions (e.g., PCA with 30 components) bridged the performance gap, achieving near-original accuracy.
- **Supervised vs. Unsupervised**:
    - LDA outperformed unsupervised methods (PCA, FastICA, NMF) in lower-dimensional spaces due to its supervised nature.
    - However, LDA's limitation to class-separable data restricts its application in general scenarios.

7. Conclusion

1. **PCA** is the most reliable unsupervised method for dimensionality reduction, balancing computational efficiency and classification performance.
2. **LDA** excels in supervised settings, reducing dimensions effectively while preserving class separability.
3. Dimensionality reduction can lead to performance trade-offs, which can be mitigated by tuning the number of components.

8. Recommendations

1. When using dimensionality reduction, consider the trade-off between computational efficiency and classification accuracy.
2. Use **PCA** for general-purpose reduction and **LDA** for supervised tasks.
3. For high-dimensional datasets, experiment with component counts to find the optimal balance.