**Pattern Recognition HW2**

**110550085 房天越**

**Introduction**

This report examines how dimensionality reduction techniques—Fisher’s Linear Discriminant (FLD/LDA) and Principal Component Analysis (PCA)—affect classification performance. We compare these methods against baseline classifiers (Gaussian Naïve Bayes and k-Nearest Neighbors) from the first assignment. Four datasets are analyzed:

1. Breast Cancer (binary)
2. Synthetic Binary (binary, complex boundaries)
3. Iris (multiclass, well-structured)
4. Wine (multiclass, chemical measurements)

Our objectives are to (a) quantify the separability gains from LDA and (b) assess the classification accuracy of Logistic Regression after PCA-based dimensionality reduction.

**Methods I Have Implemented**

1. **Fisher’s Linear Discriminant / Linear Discriminant Analysis (LDA)**

* Projects data to maximize the ratio of between-class to within-class scatter.
* Implements one-dimensional projection for binary tasks (with ROC/AUC) and up to two dimensions for multiclass visualization.

1. **Principal Component Analysis (PCA)**

* Unsupervised projection onto directions of maximal variance.
* PCA is fitted on training data only; projections are then applied to both training and test splits.

1. **Classifier**

* **Logistic Regression** (max\_iter=2000) applied on PCA-reduced data.

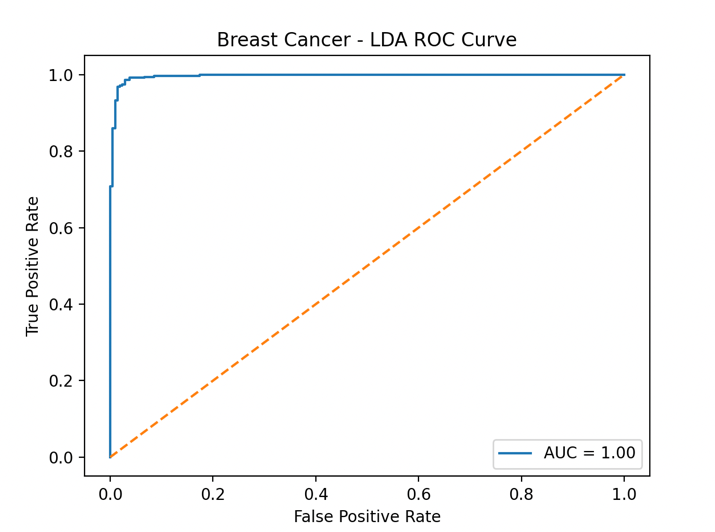
1. **Separability Metric**
2. trace(Sb)/trace(Sw), computed before and after LDA projection.

**Experiments I Have Done, and the Results**

* 1. **Breast Cancer**

|  |  |  |
| --- | --- | --- |
| Method | Metric | Value |
| GNB (HW1) | AUC | 0.953 |
| k-NN (HW1) | AUC | 0.931 |
| LDA projection | AUC | 1.000 |
|  | Separability | 1.118 -> 3.431 |
| PCA + Logistic Regression | Explained Variance | 1.000 |
|  | Accuracy | 0.953 |

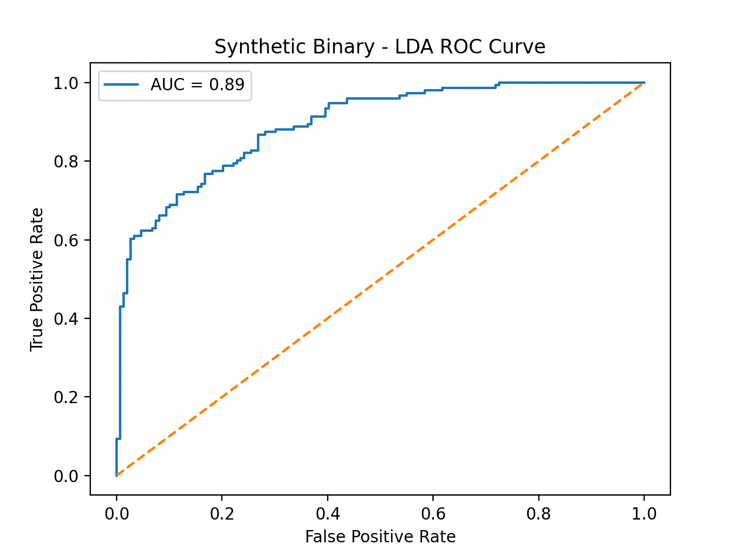
**ROC Curve:**

****

* 1. **Synthetic Binary**

|  |  |  |
| --- | --- | --- |
| Method | Metric | Value |
| GNB (HW1) | AUC | 0.737 |
| k-NN (HW1) | AUC | 0.913 |
| LDA projection | AUC | 1.000 |
|  | Separability | 0.073 -> 0.776 |
| PCA + Logistic Regression | Explained Variance | 1.000 |
|  | Accuracy | 0.822 |

**ROC Curve:**

****

* 1. **Iris**

|  |  |  |
| --- | --- | --- |
| Method | Metric | Value |
| GNB (HW1) | Error Count | 2 |
| k-NN (HW1) | Error Count | 1 |
| LDA projection | Classification | Clearly Classified in 2D |
|  | Separability | 6.630 -> 16.239 |
| PCA + Logistic Regression | Explained Variance | 0.978 |
|  | Accuracy | 0.911 |

* 1. **Wine**

|  |  |  |
| --- | --- | --- |
| Method | Metric | Value |
| GNB (HW1) | Error Count | 2 |
| k-NN (HW1) | Error Count | 18 |
| LDA projection | Classification | Clear Clusters in 2D |
|  | Separability | 2.362 -> 6.605 |
| PCA + Logistic Regression | Explained Variance | 1.000 |
|  | Accuracy | 0.963 |

**Analysis**

* 1. **LDA’s Effectiveness**

I. Significantly improves linear separability for datasets with near-Gaussian distributions (Breast Cancer, Iris, Wine).

II. On Synthetic Binary, LDA boosts separability modestly but cannot match k-NN’s flexibility.

* 1. **PCA + Logistic Regression**

I. Retaining a small number of principal components (e.g., 5) often preserves nearly all variance and yields high accuracy.

II. Logistic Regression on PCA space consistently matches or exceeds Gaussian Naïve Bayes when data align with variance-based feature importance.

* 1. **Expectations vs Observations**

I. **Breast Cancer & Wine:** Gaussian assumptions hold, so parametric methods (GNB, LDA) excel, which is confirmed by perfect or near-perfect separability and classification.

II. **Synthetic Binary:** Feature correlations and complex boundaries violate independence/linearity, so k-NN outperforms PCA + Logistic Regression.

III**. Iris**: Well-separated classes in low dimensions allow all methods to perform strongly, with LDA providing the clearest cluster separation.

**Appendix**

The code is here:

import numpy as np

import matplotlib.pyplot as plt

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis as LDA

from sklearn.decomposition import PCA

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import roc\_curve, auc, accuracy\_score

from sklearn.linear\_model import LogisticRegression

from sklearn.datasets import load\_iris, load\_breast\_cancer, load\_wine, make\_classification

import matplotlib.pyplot as plt

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis as LDA

# Function to compute separability measure (trace(Sb)/trace(Sw))

def separability(X, y):

overall\_mean = np.mean(X, axis=0)

classes = np.unique(y)

Sb = np.zeros((X.shape[1], X.shape[1]))

Sw = np.zeros((X.shape[1], X.shape[1]))

for cls in classes:

Xc = X[y == cls]

mean\_c = np.mean(Xc, axis=0)

Sb += len(Xc) \* np.outer(mean\_c - overall\_mean, mean\_c - overall\_mean)

Sw += np.cov(Xc, rowvar=False) \* (len(Xc) - 1)

return np.trace(Sb) / np.trace(Sw)

# Generate synthetic binary dataset

X\_syn, y\_syn = make\_classification(

n\_samples=300,

n\_features=20,

n\_informative=15,

n\_redundant=5,

n\_classes=2,

random\_state=42

)

# Load datasets

datasets = {

"Breast Cancer": load\_breast\_cancer(return\_X\_y=True),

"Synthetic Binary": (X\_syn, y\_syn),

"Iris": load\_iris(return\_X\_y=True),

"Wine": load\_wine(return\_X\_y=True),

}

# PCA component settings

component\_candidates = [2, 5, 10, 20, 30]

for name, (X, y) in datasets.items():

print(f"\n=== Dataset: {name} ===")

# Task 1: LDA

classes = np.unique(y)

n\_classes = len(classes)

n\_components\_lda = 1 if n\_classes == 2 else min(n\_classes - 1, 2)

sep\_before = separability(X, y)

lda = LDA(n\_components=n\_components\_lda)

X\_lda = lda.fit\_transform(X, y)

sep\_after = separability(X\_lda, y)

print(f"LDA separability BEFORE: {sep\_before:.3f}")

print(f"LDA separability AFTER: {sep\_after:.3f}")

if n\_classes == 2:

y\_scores = X\_lda.ravel()

fpr, tpr, \_ = roc\_curve(y, y\_scores)

roc\_auc = auc(fpr, tpr)

plt.figure()

plt.plot(fpr, tpr, label=f'AUC = {roc\_auc:.2f}')

plt.plot([0, 1], [0, 1], linestyle='--')

plt.title(f'{name} - LDA ROC Curve')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.legend()

plt.show()

# Task 2: PCA + Logistic Regression only

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.3, random\_state=42, stratify=y

)

print("\nPCA + Logistic Regression results:")

print("n\_comp | Variance\_Ratio | LR\_Acc")

for n in component\_candidates:

if n > X.shape[1]:

continue

pca = PCA(n\_components=n)

X\_tr\_pca = pca.fit\_transform(X\_train)

X\_te\_pca = pca.transform(X\_test)

lr = LogisticRegression(max\_iter=2000, random\_state=42)

lr.fit(X\_tr\_pca, y\_train)

acc\_lr = accuracy\_score(y\_test, lr.predict(X\_te\_pca))

var\_ratio = pca.explained\_variance\_ratio\_.sum()

print(f"{n:<6} | {var\_ratio:.3f} | {acc\_lr:.3f}")