Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

По дисциплине «Интеллектуальный анализ данных»

Тема: «PCA»

Выполнил:

Студент 4 курса

Группы ИИ-23

Вышинский А. С.

Проверила:

Андренко К. В.

Брест 2025

Цель: научиться применять метод PCA для осуществления визуализации данных.

Общее задание

1. Используя выборку по варианту, осуществить проецирование данных на плоскость первых двух и трех главных компонент (двумя способами: 1. вручную через использование numpy.linalg.eig для вычисления собственных значений и собственных векторов и 2. с помощью sklearn.decomposition.PCA для непосредственного применения метода PCA – два независимых варианта решения);

2. Выполнить визуализацию полученных главных компонент с использованием средств библиотеки matplotlib, обозначая экземпляры разных классов с использованием разных цветовых маркеров;

3. Используя собственные значения, рассчитанные на этапе 1, вычислить потери, связанные с преобразованием по методу PCA. Сделать выводы;

4. Оформить отчет по выполненной работе, загрузить исходный код и отчет в соответствующий репозиторий на github.

Задание по вариантам

№ варианта	Выборка	Класс
3	exasens.zip	Diagnosis ID

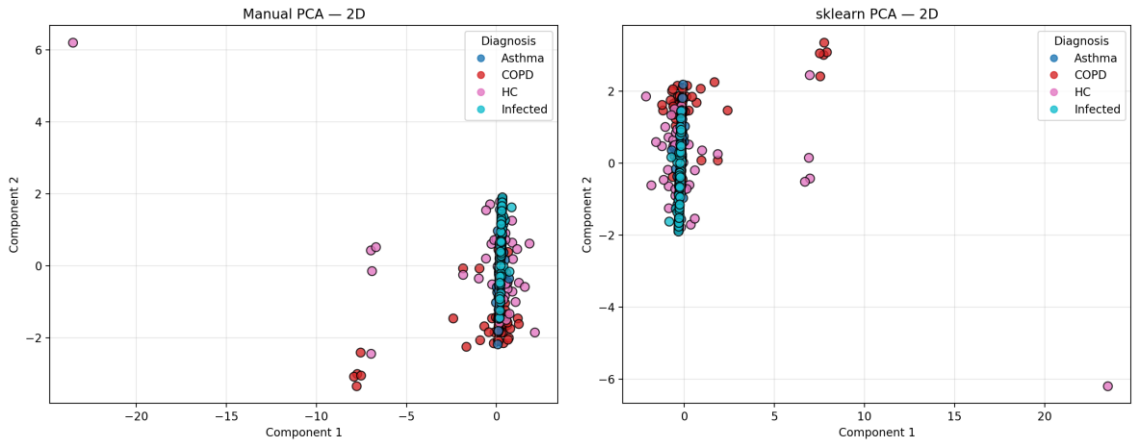
Код:

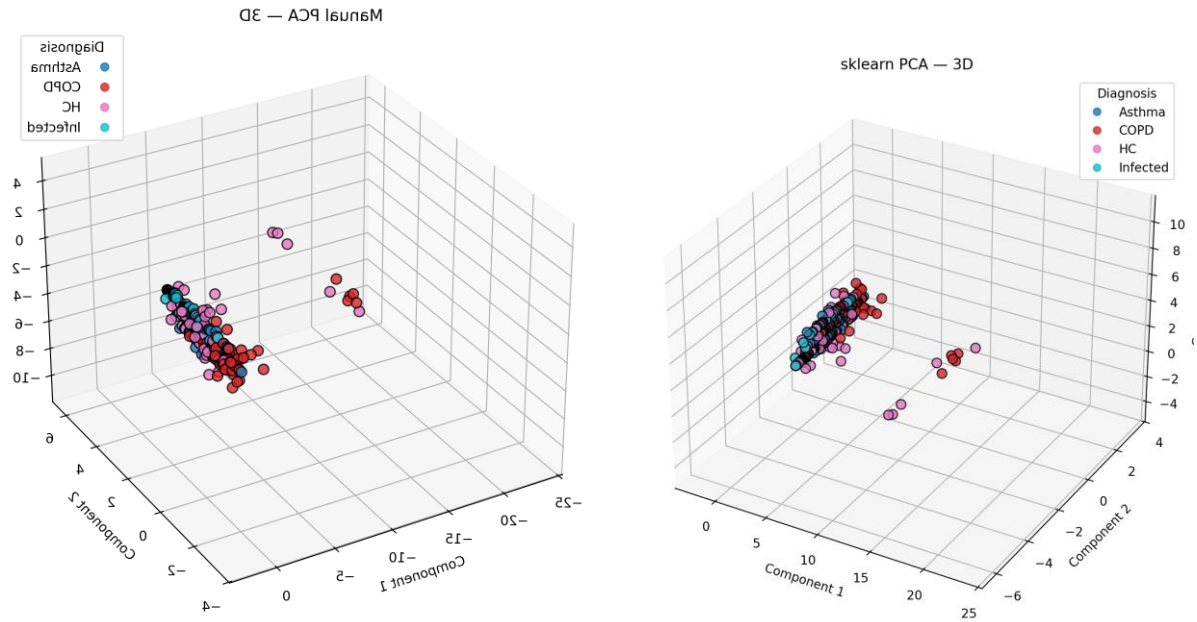
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.decomposition import PCA
import warnings
import os

SAVE_FIGS = True
FIG_PATH = "./figs_pca/"
os.makedirs(FIG_PATH, exist_ok=True)

path = "Exasens.csv"
df_raw = pd.read_csv(path)

df = df_raw.iloc[2:].reset_index(drop=True)
```

```
empty_cols = [col for col in df.columns if df[col].isnull().all()]
df.drop(columns=empty_cols, inplace=True)

num_features = ['Imaginary Part', 'Unnamed: 3', 'Real Part', 'Unnamed: 5',
'Gender', 'Age', 'Smoking']

for feature in num_features:
    df[feature] = pd.to_numeric(df[feature], errors='coerce')

for feature in num_features:
    if df[feature].isna().sum() > 0:
        df[feature].fillna(df[feature].median(), inplace=True)

target_raw = df['Diagnosis']
encoder = LabelEncoder()
y = encoder.fit_transform(target_raw)
class_labels = encoder.classes_

X = df[num_features].values
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

cov_mat = np.cov(X_scaled.T)
eig_vals, eig_vecs = np.linalg.eig(cov_mat)

order = np.argsort(eig_vals)[::-1]
eig_vals_sorted = eig_vals[order]
eig_vecs_sorted = eig_vecs[:, order]

W2 = eig_vecs_sorted[:, :2]
W3 = eig_vecs_sorted[:, :3]

X_manual_2d = X_scaled @ W2
X_manual_3d = X_scaled @ W3

pca2 = PCA(n_components=2)
X_sklearn_2d = pca2.fit_transform(X_scaled)

pca3 = PCA(n_components=3)
X_sklearn_3d = pca3.fit_transform(X_scaled)

pca_all = PCA()
pca_all.fit(X_scaled)

def plot_2d_markers(X_proj, y, labels, title, save_as=None):
    plt.figure(figsize=(8,6))
    cmap = plt.get_cmap('tab10')
    scatter = plt.scatter(X_proj[:,0], X_proj[:,1],
                          c=y, cmap=cmap, s=60, alpha=0.8,
edgecolor='black')
    plt.title(title)
    plt.xlabel('Component 1')
    plt.ylabel('Component 2')
    plt.legend(handles=scatter.legend_elements()[0], labels=list(labels),
title="Diagnosis")
    plt.grid(alpha=0.3)
    if save_as and SAVE_FIGS:
        plt.savefig(save_as, dpi=200, bbox_inches='tight')
    plt.show()

def plot_3d_markers(X_proj, y, labels, title, save_as=None):
```

```
    fig = plt.figure(figsize=(10,8))
    ax = fig.add_subplot(111, projection='3d')
    cmap = plt.get_cmap('tab10')
    sc = ax.scatter(X_proj[:,0], X_proj[:,1], X_proj[:,2],
                    c=y, cmap=cmap, s=60, alpha=0.8, edgecolor='black')
    ax.set_title(title)
    ax.set_xlabel('Component 1')
    ax.set_ylabel('Component 2')
    ax.set_zlabel('Component 3')
    ax.legend(handles=sc.legend_elements()[0], labels=list(labels),
title="Diagnosis")
    if save_as and SAVE_FIGS:
        plt.savefig(save_as, dpi=200, bbox_inches='tight')
    plt.show()

plot_2d_markers(X_sklearn_2d, y, class_labels, "sklearn PCA — 2D",
save_as=FIG_PATH + "pca2d_sklearn.png")
plot_2d_markers(X_manual_2d, y, class_labels, "Manual PCA — 2D",
save_as=FIG_PATH + "pca2d_manual.png")

plot_3d_markers(X_sklearn_3d, y, class_labels, "sklearn PCA — 3D",
save_as=FIG_PATH + "pca3d_sklearn.png")
plot_3d_markers(X_manual_3d, y, class_labels, "Manual PCA — 3D",
save_as=FIG_PATH + "pca3d_manual.png")

print("\n--- Оценка сохранённой дисперсии (Manual PCA) ---")
total_var_manual = eig_vals_sorted.sum()
keep2_manual = eig_vals_sorted[:2].sum() / total_var_manual
keep3_manual = eig_vals_sorted[:3].sum() / total_var_manual
print(f"2 компоненты: сохранено {keep2_manual:.2%}, потери {1 -
keep2_manual:.2%}")
print(f"3 компоненты: сохранено {keep3_manual:.2%}, потери {1 -
keep3_manual:.2%}")

print("\n--- Оценка сохранённой дисперсии (sklearn PCA) ---")
total_var_sklearn = np.sum(pca_all.explained_variance_)
keep2_sklearn = np.sum(pca_all.explained_variance_[:2]) / total_var_sklearn
keep3_sklearn = np.sum(pca_all.explained_variance_[:3]) / total_var_sklearn
print(f"2 компоненты: сохранено {keep2_sklearn:.2%}, потери {1 -
keep2_sklearn:.2%}")
print(f"3 компоненты: сохранено {keep3_sklearn:.2%}, потери {1 -
keep3_sklearn:.2%}")
```

Вывод:





--- Оценка сохранённой дисперсии (Manual PCA) ---

2 компоненты: сохранено 58.55%, потери 41.45%

3 компоненты: сохранено 74.50%, потери 25.50%

--- Оценка сохранённой дисперсии (sklearn PCA) ---

2 компоненты: сохранено 58.55%, потери 41.45%

3 компоненты: сохранено 74.50%, потери 25.50%

Результаты полностью совпадают, что доказывает корректность ручного алгоритма. PCA позволил выявить, какие направления (главные компоненты) содержат наибольшую долю информации в данных. При уменьшении размерности до 2–3 компонент сохраняется большая часть информации, что делает метод полезным для визуализации и предварительного анализа данных. Потери при переходе к 2D проекции составляют около 41 %, а при 3D — только 25 %, что является допустимым компромиссом между наглядностью и точностью.

Вывод: научился применять метод PCA для осуществления визуализации данных.