

**Министерство образования Республики Беларусь**  
**Учреждение образования**  
**«Брестский государственный технический**  
**университет»**  
Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе № 1  
«РСА»

**Выполнил:**  
Студент  
группы ИИ-22  
Сидоренко А.А.

**Проверил:**  
Крощенко А.А.

Брест 2024

**Цель:** научиться применять метод PCA для осуществления визуализации данных.

**Постановка задачи:**

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 вывод

**Ход работы:**

Вариант 16

**Код:**

```
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

data = pd.read_csv("seeds_dataset.txt", delim_whitespace=True, header=None)
feature = data.iloc[:, :-1].values
target = data.iloc[:, -1].values

scaler = StandardScaler()
feature_scd = scaler.fit_transform(feature)

#2 comps
pca2 = PCA(n_components=2)
pca_2cmp = pca2.fit_transform(feature_scd)
loss2 = 1 - np.sum(pca2.explained_variance_ratio_)

#show res 2 comps
plt.figure(figsize=(10, 6))

for label in np.unique(target):
    plt.scatter(pca_2cmp[target == label, 0], pca_2cmp[target == label, 1], label=f"Class {label}")

plt.xlabel("Comp 1")
plt.ylabel("Comp 2")
plt.title("PCA 2")
plt.legend()
plt.show()

#3 comps
pca3 = PCA(n_components=3)
pca_3cmp = pca3.fit_transform(feature_scd)
loss3 = 1 - np.sum(pca3.explained_variance_ratio_)

#show res 3 comps
plot = plt.figure(figsize=(10, 6))
z = plot.add_subplot(111, projection='3d')

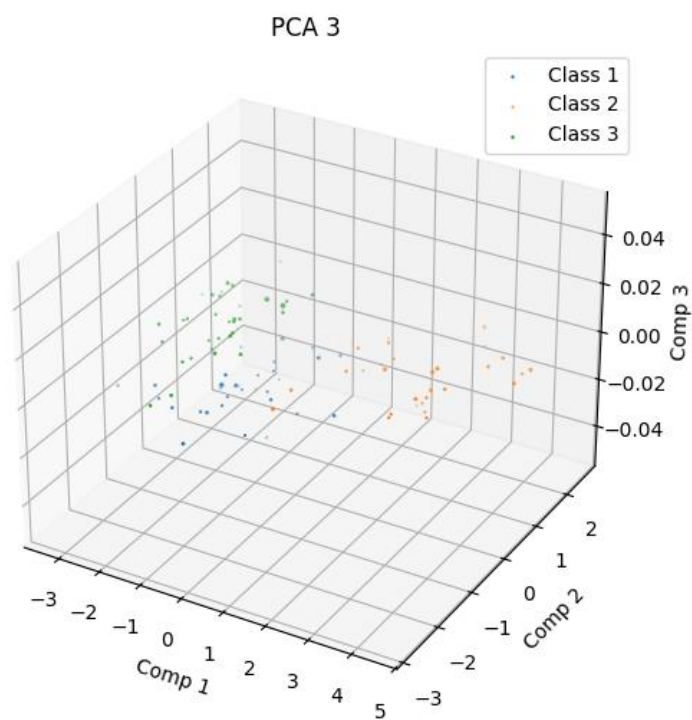
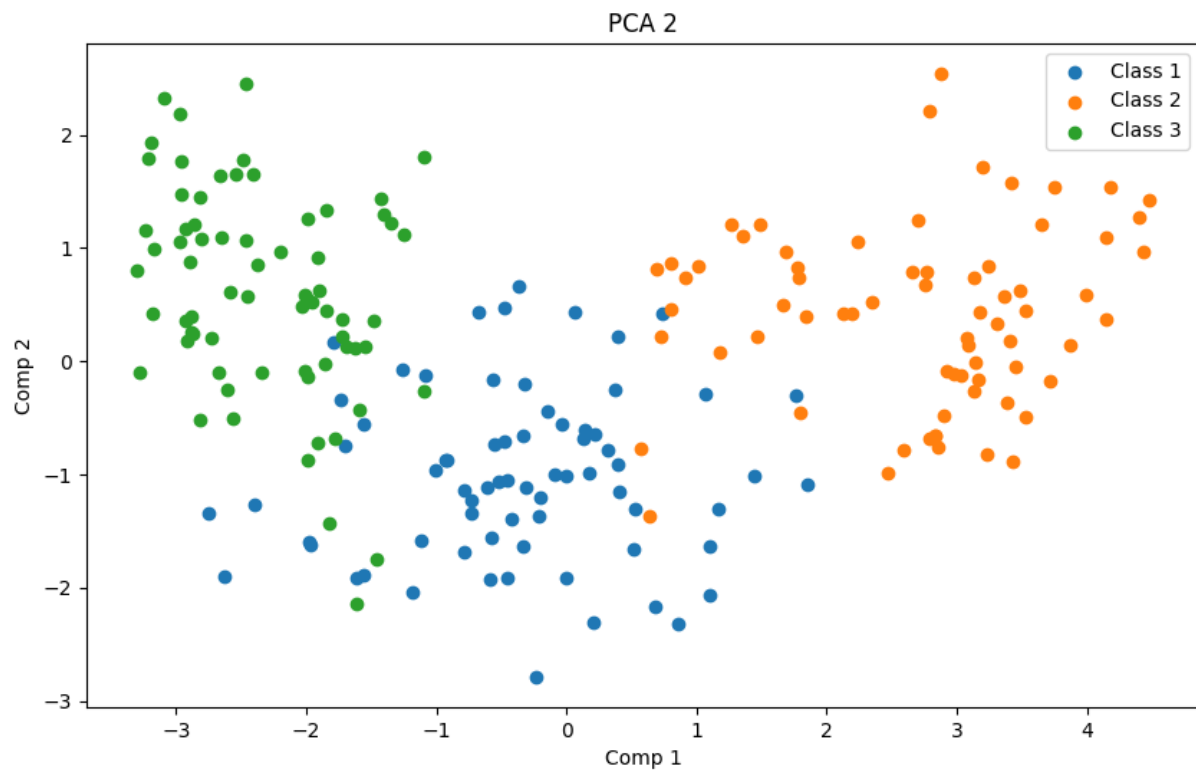
for label in np.unique(target):
    plt.scatter(pca_3cmp[target == label, 0], pca_3cmp[target == label, 1], pca_3cmp[target == label, 2], label=f"Class {label}")

plt.xlabel("Comp 1")
```

```
plt.ylabel("Comp 2")
z.set_zlabel("Comp 3")
plt.title("PCA 3")
plt.legend()
plt.show()

print(f'потери при 2 comps: {loss2}')
print(f'потери при 3 comps: {loss3}')
```

## Результат:



**Потери при 2 компонентах: 0.1101751381508762**  
**Потери при 3 компонентах: 0.013317504067955288**

**Вывод:** научился применять метод PCA для осуществления визуализации данных, потери данных при существенном сжатии являются не значительными.