

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №1

Специальность ИИ-23

Выполнил

А.П. Романюк,

студент группы ИИ-23

Проверила

К.В. Андренко,

«—» ————— 2025 г.

Цель работы: 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 варианту:

№ варианта	Выборка	Класс
9	heart+failure+clinical+records.zip	death_event

Ход работы:

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

raw_data = pd.read_csv('data.csv')
raw_data.head()

labels = raw_data['DEATH_EVENT']
data = raw_data.drop(columns=['DEATH_EVENT'])
data

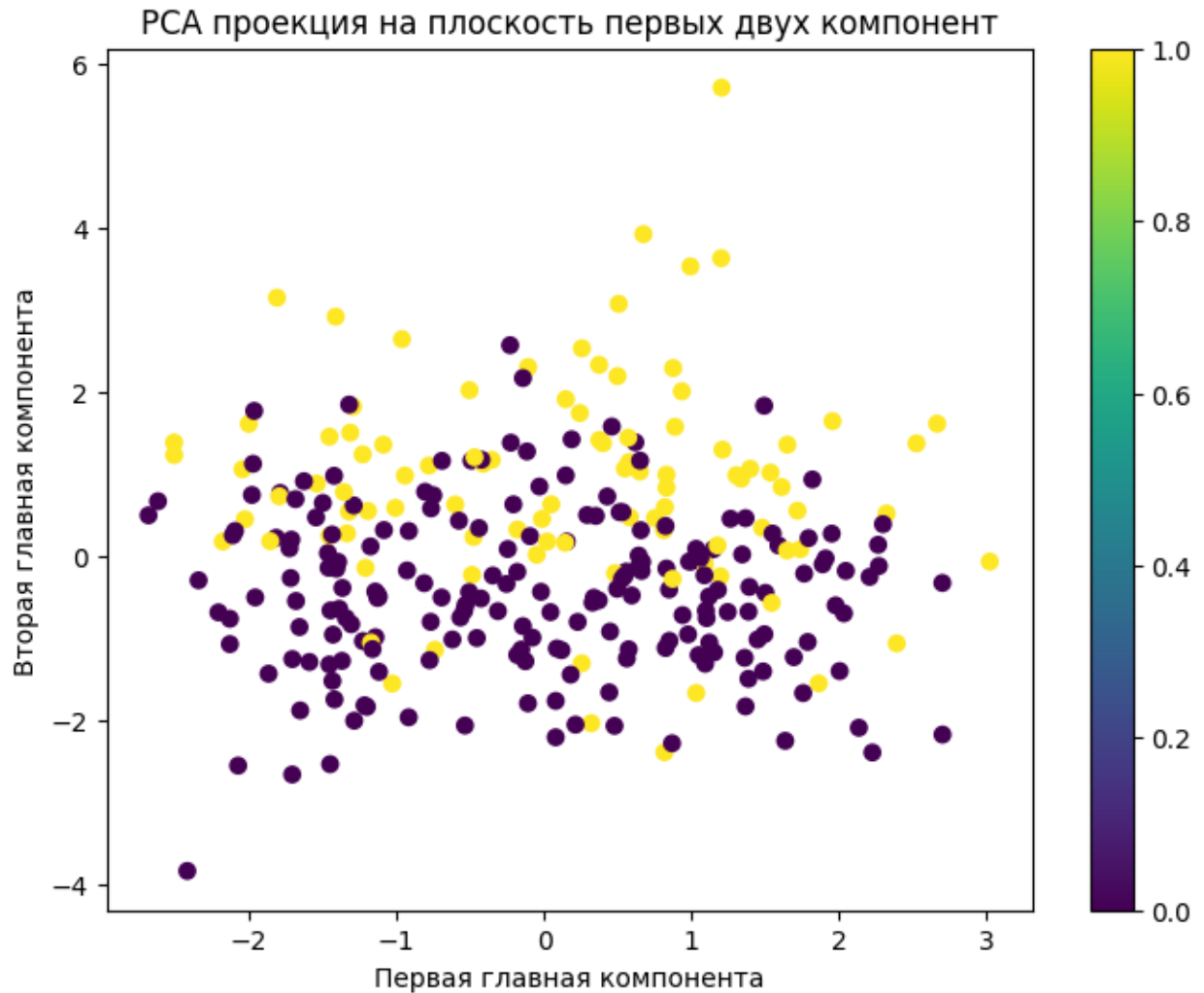
scaler = StandardScaler()
data = scaler.fit_transform(data)
data

data_centred = data - data.mean()
cov_matrix = np.cov(data_centred, rowvar=False)
eig_values, eig_vectors = np.linalg.eig(cov_matrix)
idx = np.argsort(eig_values)[-1]
eig_vectors = eig_vectors[:,idx]

data_2d = data_centred.dot(eig_vectors[:,2])
data_3d = data_centred.dot(eig_vectors[:,3])

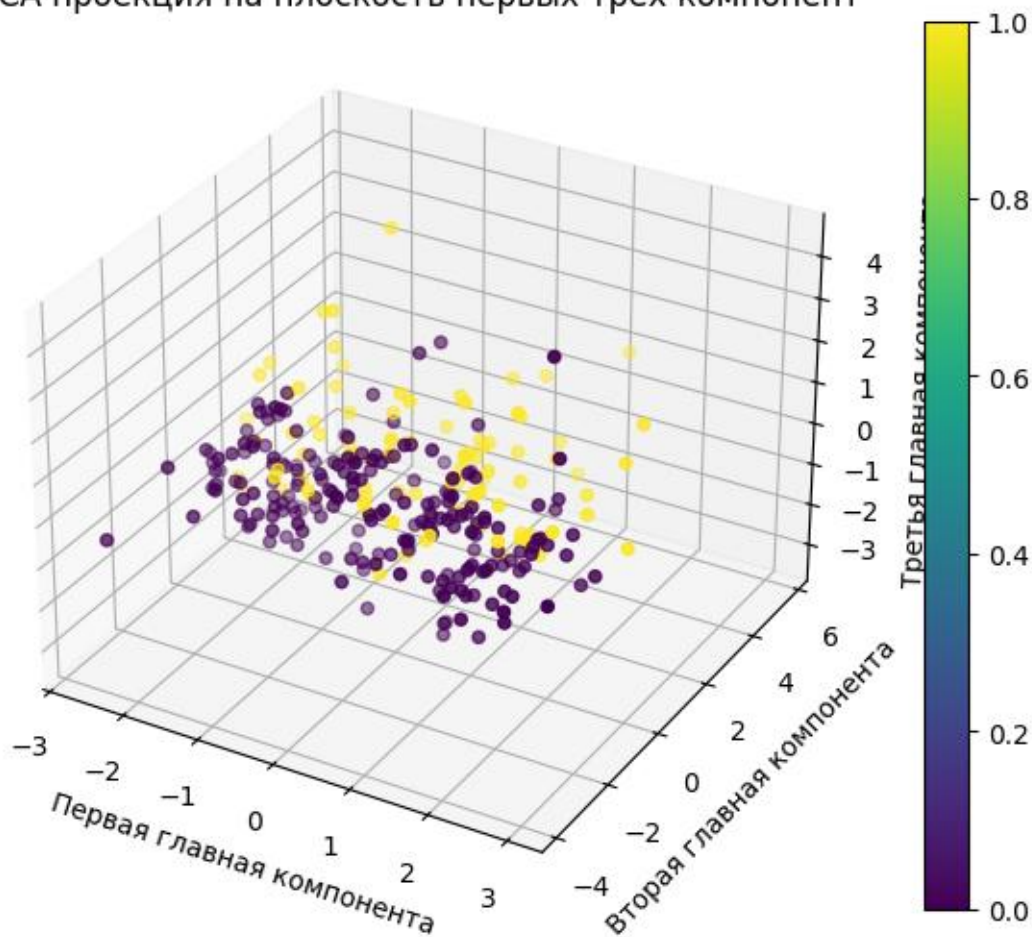
plt.figure(figsize=(8, 6))
scatter = plt.scatter(data_2d[:, 0], data_2d[:, 1], c=labels, cmap='viridis')
plt.colorbar(scatter)
```

```
plt.xlabel('Первая главная компонента')
plt.ylabel('Вторая главная компонента')
plt.title('PCA проекция на плоскость первых двух компонент')
plt.show()
```



```
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')
scatter = ax.scatter(data_3d[:, 0], data_3d[:, 1], data_3d[:, 2], c=labels, cmap='viridis')
fig.colorbar(scatter)
ax.set_xlabel('Первая главная компонента')
ax.set_ylabel('Вторая главная компонента')
ax.set_zlabel('Третья главная компонента')
ax.set_title('PCA проекция на плоскость первых трёх компонент')
plt.show()
```

PCA проекция на плоскость первых трёх компонент

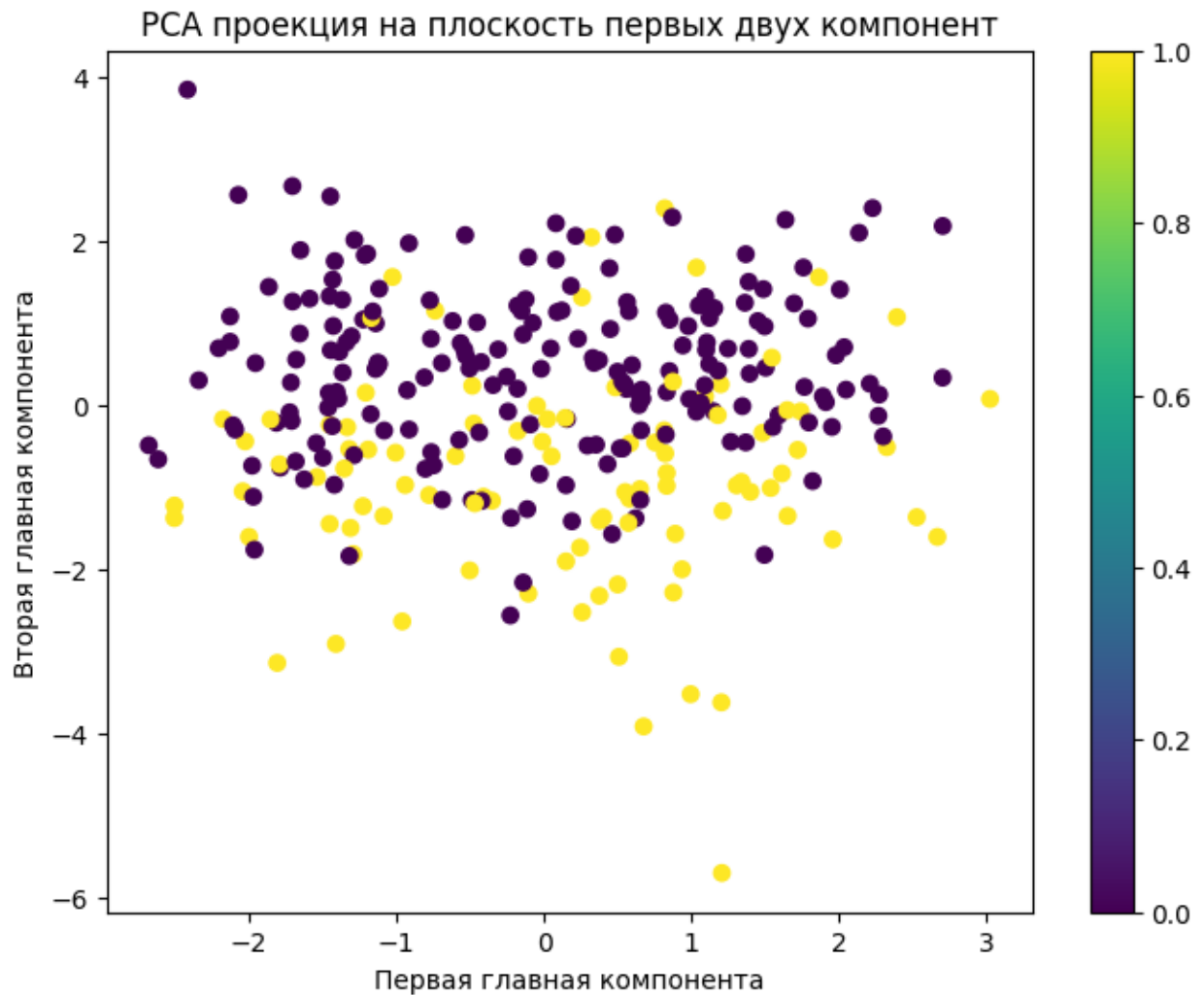


```
from sklearn.decomposition import PCA
```

```
pca_2d = PCA(n_components=2)  
data_2d = pca_2d.fit_transform(data)
```

```
pca_3d = PCA(n_components=3)  
data_3d = pca_3d.fit_transform(data)
```

```
plt.figure(figsize=(8, 6))  
scatter = plt.scatter(data_2d[:, 0], data_2d[:, 1], c=labels, cmap='viridis')  
plt.colorbar(scatter)  
plt.xlabel('Первая главная компонента')  
plt.ylabel('Вторая главная компонента')  
plt.title('PCA проекция на плоскость первых двух компонент')  
plt.show()
```



```
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')
scatter = ax.scatter(data_3d[:, 0], data_3d[:, 1], data_3d[:, 2], c=labels, cmap='viridis')
fig.colorbar(scatter)
ax.set_xlabel('Первая главная компонента')
ax.set_ylabel('Вторая главная компонента')
ax.set_zlabel('Третья главная компонента')
ax.set_title('PCA проекция на плоскость первых трёх компонент')
plt.show()
```

```
total_var = np.sum(eig_values)
```

```
explained_var_ratio = np.cumsum(sorted(eig_values, reverse=True)) / total_var
```

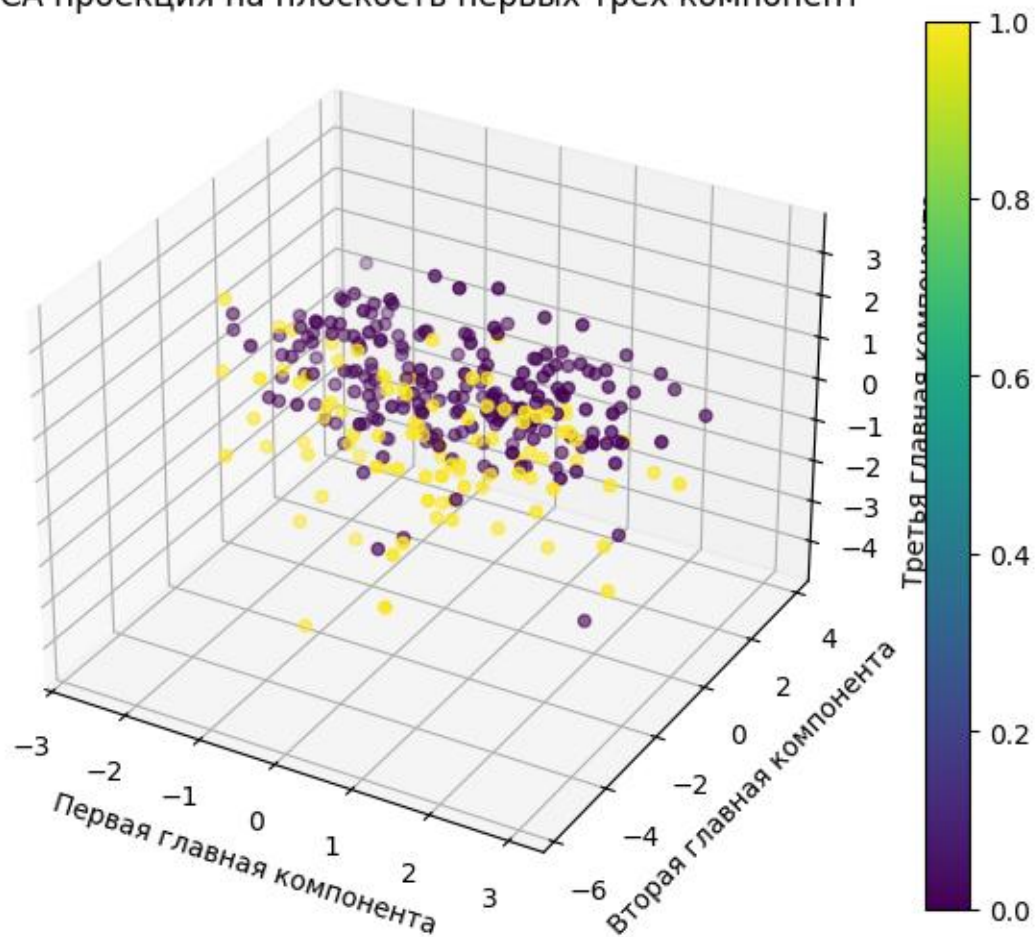
```
loss_2d = 1 - explained_var_ratio[1]
```

```
loss_3d = 1 - explained_var_ratio[2]
```

```
print("Потери при 2D проекции:", loss_2d)
```

```
print("Потери при 3D проекции:", loss_3d)
```

РСА проекция на плоскость первых трёх компонент



Вывод: научился применять метод РСА для осуществления визуализации данных