`

Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

Отчет по лабораторной работе 2

Специальность ИИ-23

Выполнил:

Макаревич Н.Р.

Студент группы ИИ-23

Проверил:

Андренко К. В.
Преподаватель-стажёр
Кафедры ИИТ,
«___» _____ 2025 г.

Брест 2025

`

Лабораторная работа № 2. Автоэнкодеры

Цель: научиться применять автоэнкодеры для осуществления визуализации данных и их анализа

Общее задание

1. Используя выборку по варианту, осуществить проецирование данных на плоскость первых двух и трех главных компонент с использованием нейросетевой модели автоэнкодера (с двумя и тремя нейронами в среднем слое);

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

3. Реализовать метод t-SNE для визуализации данных (использовать также 2 и 3 компонента), построить соответствующую визуализацию;

4. Применить к данным метод PCA (2 и 3 компонента), реализованный в ЛР №1, сделать выводы;

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

Код программы:

```
# -*- coding: utf-8 -*-

"""iad_2.ipynb


Automatically generated by Colab.


Original file is located at

    https://colab.research.google.com/drive/1BpEHLdgaiF2gwtGcvSnVMc_JT13GtbUu
"""


pip install ucimlrepo
```

`

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
from sklearn.manifold import TSNE
from sklearn.decomposition import PCA
from ucimlrepo import fetch_ucirepo
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.optimizers import Adam

from ucimlrepo import fetch_ucirepo

rice_cammeo_and_osmancik = fetch_ucirepo(id=545)

X = rice_cammeo_and_osmancik.data.features
y = rice_cammeo_and_osmancik.data.targets

print(rice_cammeo_and_osmancik.metadata)

print(rice_cammeo_and_osmancik.variables)

rice_cammeo_and_osmancik = fetch_ucirepo(id=545)
```

```
X = rice_cammeo_and_osmancik.data.features
y = rice_cammeo_and_osmancik.data.targets

y = y['Class'].map({'Cammeo': 0, 'Osmancik': 1})

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

print("Размерность X:", X_scaled.shape)
X.head()

input_dim = X_scaled.shape[1]
encoding_dim = 2

input_layer = Input(shape=(input_dim,))
encoded = Dense(6, activation='relu')(input_layer)
bottleneck = Dense(encoding_dim, activation='linear')(encoded)
decoded = Dense(6, activation='relu')(bottleneck)
output_layer = Dense(input_dim, activation='linear')(decoded)

autoencoder_2d = Model(inputs=input_layer, outputs=output_layer)
encoder_2d = Model(inputs=input_layer, outputs=bottleneck)

autoencoder_2d.compile(optimizer=Adam(learning_rate=0.01), loss='mse')

history_2d = autoencoder_2d.fit(X_scaled, X_scaled,
```

```
                    epochs=100,
                    batch_size=64,
                    verbose=0)

X_encoded_2d = encoder_2d.predict(X_scaled)

plt.figure(figsize=(8, 6))
plt.scatter(X_encoded_2d[:, 0], X_encoded_2d[:, 1], c=y, cmap='viridis', alpha=0.7)
plt.title('Автоэнкодер: 2 главные компоненты')
plt.xlabel('Компонента 1')
plt.ylabel('Компонента 2')
plt.colorbar(label='Класс')
plt.show()

encoding_dim = 3

input_layer = Input(shape=(input_dim,))
encoded = Dense(6, activation='relu')(input_layer)
bottleneck = Dense(encoding_dim, activation='linear')(encoded)
decoded = Dense(6, activation='relu')(bottleneck)
output_layer = Dense(input_dim, activation='linear')(decoded)

autoencoder_3d = Model(inputs=input_layer, outputs=output_layer)
encoder_3d = Model(inputs=input_layer, outputs=bottleneck)

autoencoder_3d.compile(optimizer=Adam(learning_rate=0.01), loss='mse')
```

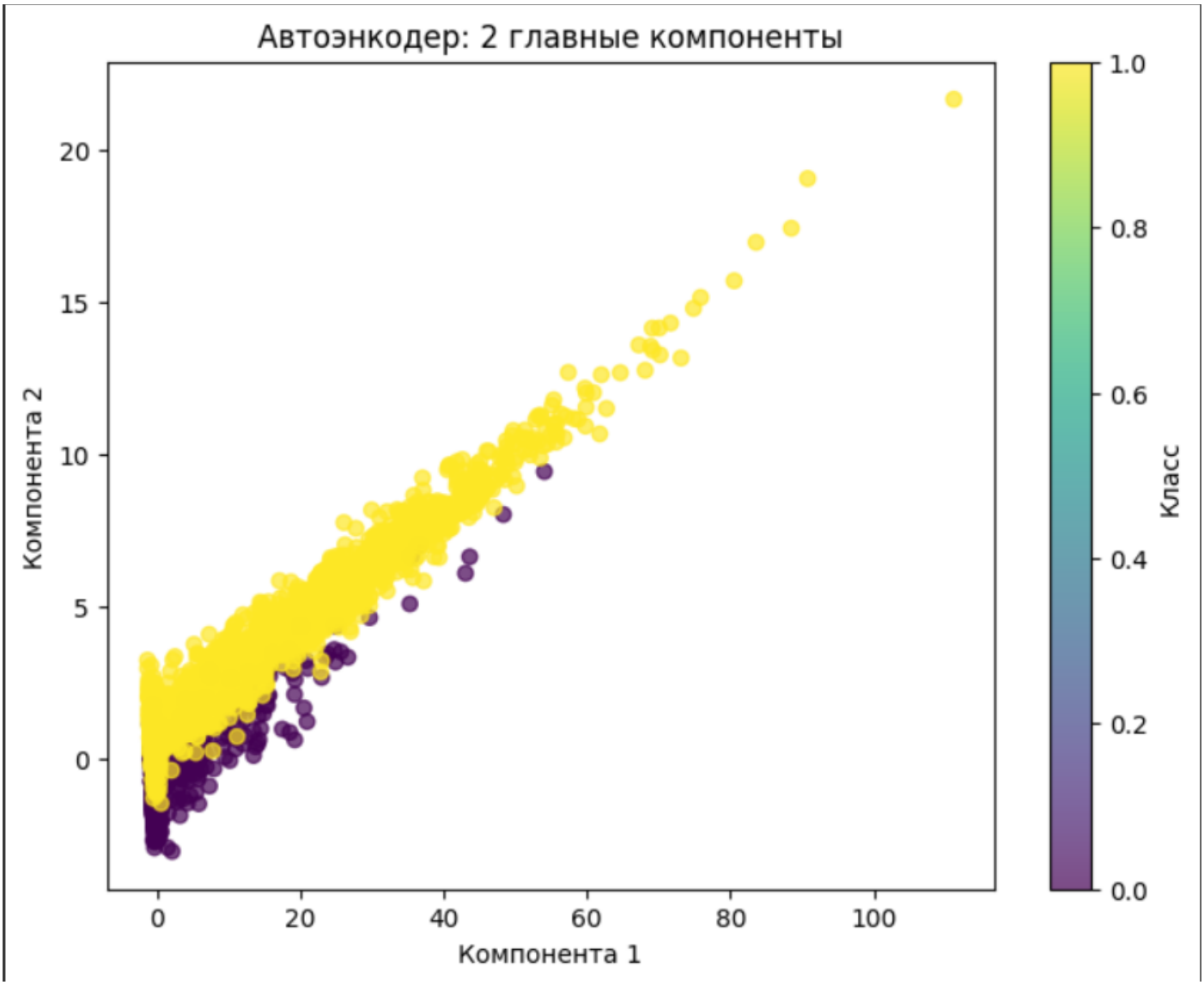
`

```
history_3d = autoencoder_3d.fit(X_scaled, X_scaled,
                                epochs=100,
                                batch_size=64,
                                verbose=0)

X_encoded_3d = encoder_3d.predict(X_scaled)

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
sc = ax.scatter(X_encoded_3d[:, 0], X_encoded_3d[:, 1], X_encoded_3d[:, 2],
                c=y, cmap='viridis', alpha=0.7)
ax.set_title('Автоэнкодер: 3 главные компоненты')
ax.set_xlabel('Компонента 1')
ax.set_ylabel('Компонента 2')
ax.set_zlabel('Компонента 3')
fig.colorbar(sc, label='Класс')
plt.show()

tsne_2d = TSNE(n_components=2, random_state=42, perplexity=30)
X_tsne_2d = tsne_2d.fit_transform(X_scaled)

plt.figure(figsize=(8, 6))
plt.scatter(X_tsne_2d[:, 0], X_tsne_2d[:, 1], c=y, cmap='viridis', alpha=0.7)
plt.title('t-SNE: 2 компоненты')
plt.xlabel('Компонента 1')
```

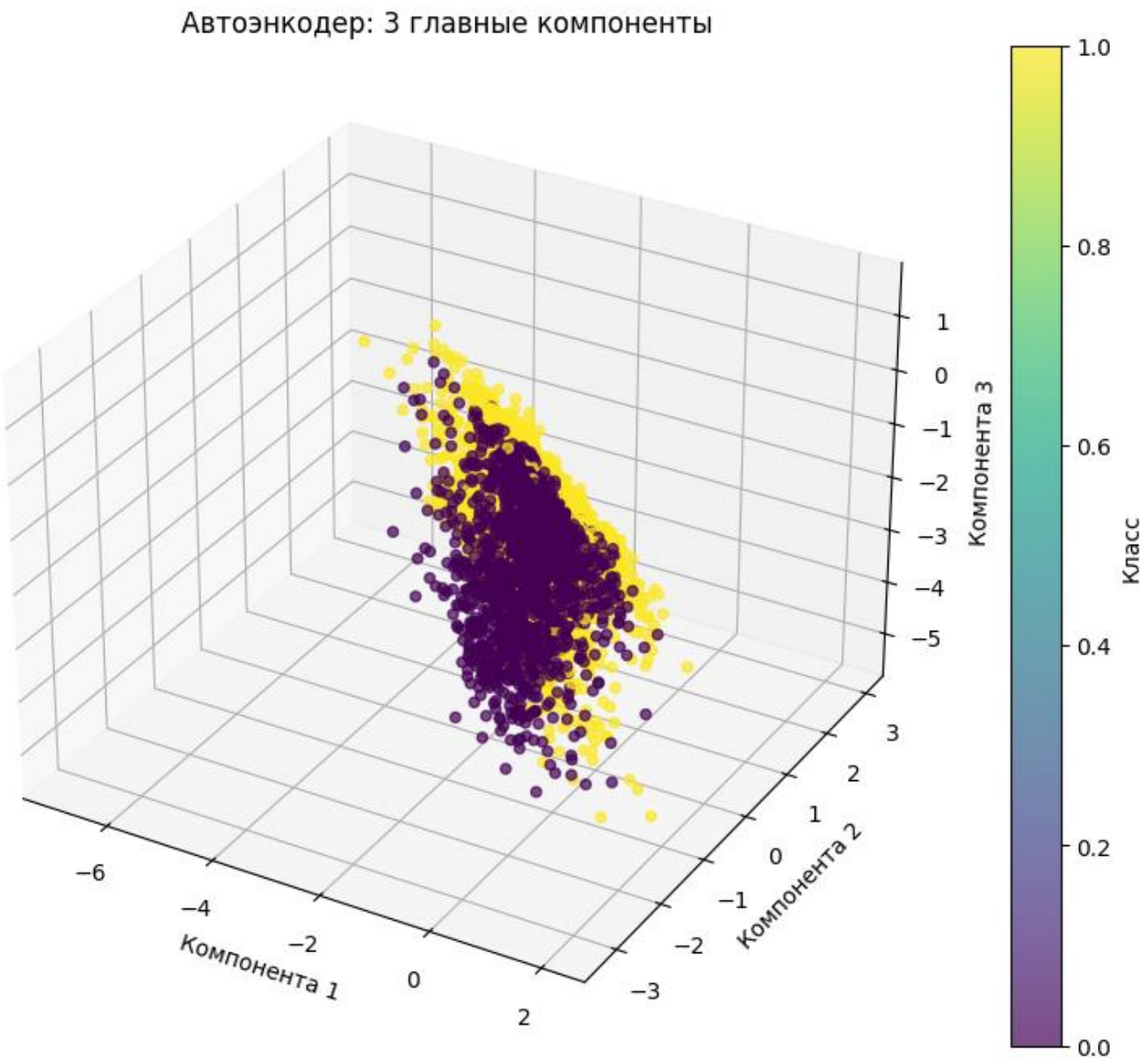
```
plt.ylabel('Компонента 2')
plt.colorbar(label='Класс')
plt.show()

tsne_3d = TSNE(n_components=3, random_state=42, perplexity=30)
X_tsne_3d = tsne_3d.fit_transform(X_scaled)

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
sc = ax.scatter(X_tsne_3d[:, 0], X_tsne_3d[:, 1], X_tsne_3d[:, 2],
                c=y, cmap='viridis', alpha=0.7)
ax.set_title('t-SNE: 3 компоненты')
ax.set_xlabel('Компонента 1')
ax.set_ylabel('Компонента 2')
ax.set_zlabel('Компонента 3')
fig.colorbar(sc, label='Класс')
plt.show()

pca_2d = PCA(n_components=2)
X_pca_2d = pca_2d.fit_transform(X_scaled)

plt.figure(figsize=(8, 6))
plt.scatter(X_pca_2d[:, 0], X_pca_2d[:, 1], c=y, cmap='viridis', alpha=0.7)
plt.title('PCA: 2 компоненты')
plt.xlabel('Компонента 1')
plt.ylabel('Компонента 2')
```

```
plt.colorbar(label='Класс')
plt.show()

pca_3d = PCA(n_components=3)
X_pca_3d = pca_3d.fit_transform(X_scaled)

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
sc = ax.scatter(X_pca_3d[:, 0], X_pca_3d[:, 1], X_pca_3d[:, 2],
                c=y, cmap='viridis', alpha=0.7)
ax.set_title('PCA: 3 компоненты')
ax.set_xlabel('Компонента 1')
ax.set_ylabel('Компонента 2')
ax.set_zlabel('Компонента 3')
fig.colorbar(sc, label='Класс')
plt.show()
```

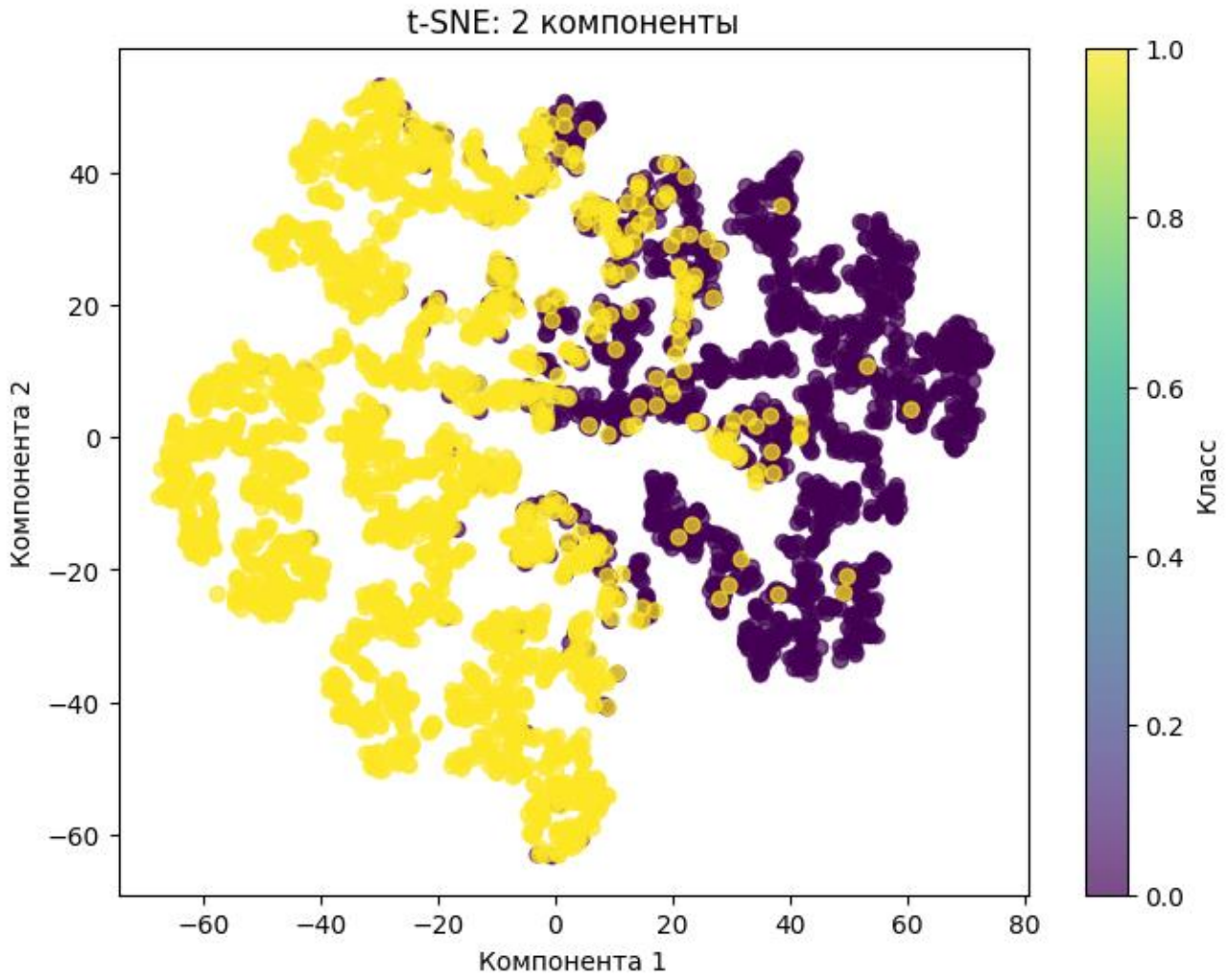
`


Автоэнкодер: 2 главные компоненты
Компонента 2
Компонента 1
Класс
0
5
10
15
20
0
20
40
60
80
100
0.0
0.2
0.4
0.6
0.8
1.0

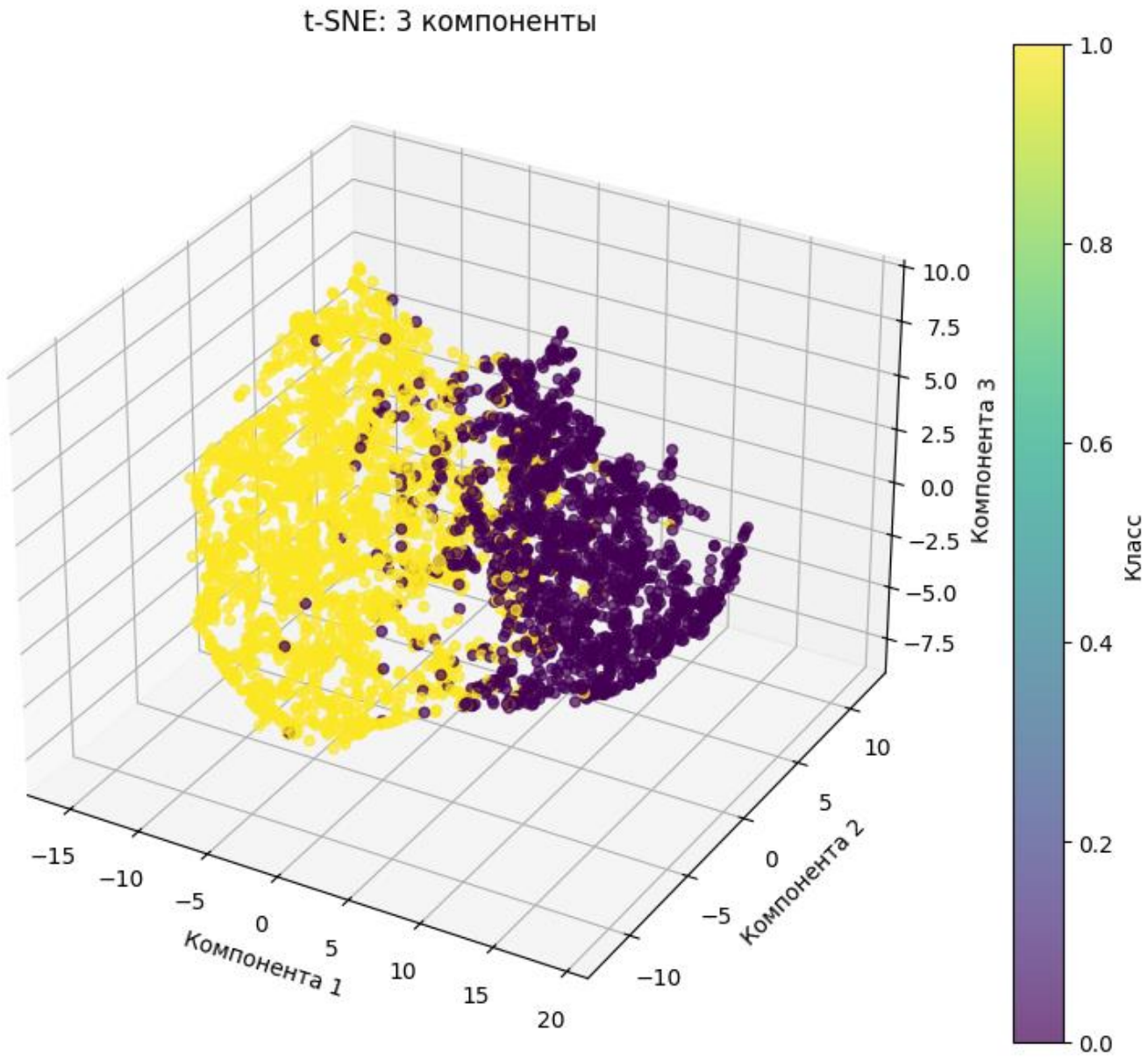
`


Автоэнкодер: 3 главные компоненты
Компонента 1
Компонента 2
Компонента 3
Класс
−6
−4
−2
0
2
−3
−2
−1
0
1
2
3
1
0
−1
−2
−3
−4
−5
1.0
0.8
0.6
0.4
0.2
0.0

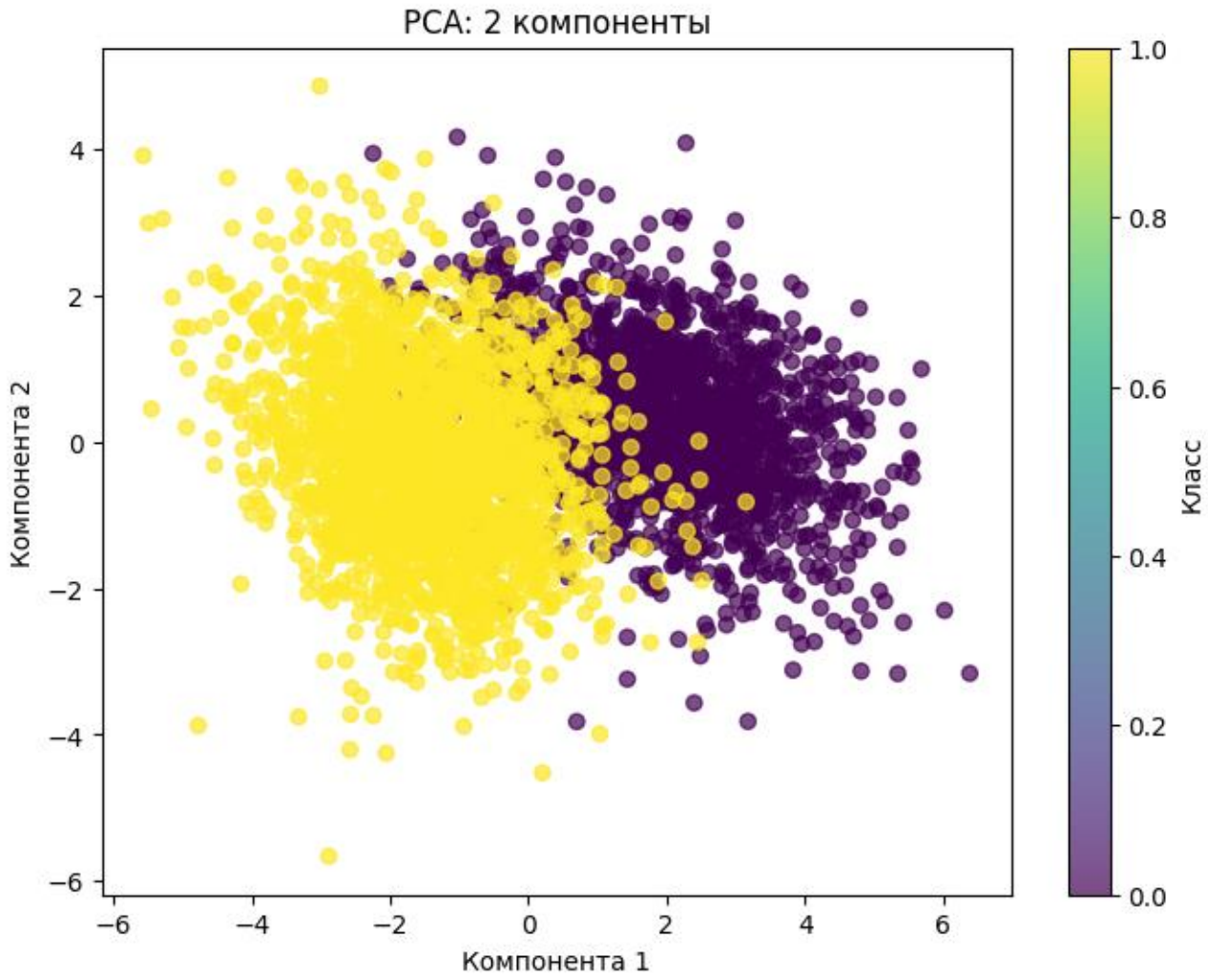
`


t-SNE: 2 компоненты
Компонента 2
Компонента 1
Класс
40
20
0
−20
−40
−60
−60
−40
−20
0
20
40
60
80
1.0
0.8
0.6
0.4
0.2
0.0

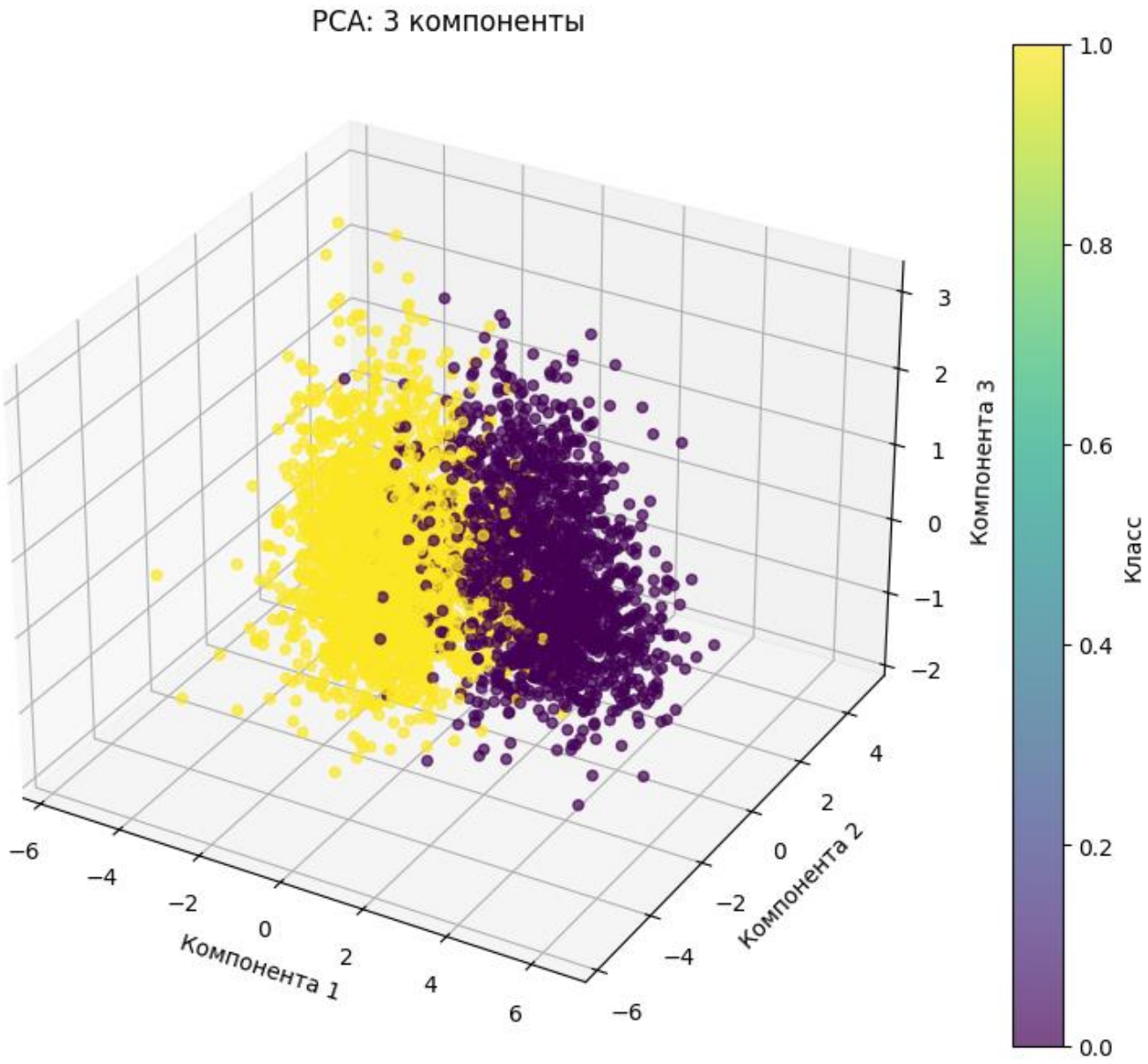
`


t-SNE: 3 компоненты
Компонента 1
Компонента 2
Компонента 3
Класс
−15
−10
−5
0
5
10
15
20
−10
−5
0
5
10
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
1.0
0.8
0.6
0.4
0.2
0.0

`


PCA: 2 компоненты
Компонента 2
Компонента 1
Класс
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
1.0
0.8
0.6
0.4
0.2
0.0

`



Вывод: научился применять автоенкодеры для визуализации данных для последующего анализа.