Ввод [9]:

```
import numpy as np
from sklearn.manifold import TSNE
from sklearn.datasets import load_iris
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import umap.umap_ as umap
```

Авторы TSNE о выборе распределения Коши в отображаемом пространстве:

> We use a Student t-distribution with a single degree of freedom, because it has the particularly nice property that $\left(1+\|y_i-y_j\|^2\right)^{-1}$ approaches an inverse square law for large pairwise distances $\|y_i-y_j\|$ in the low-dimensional map. This makes the map's representation of joint probabilities (almost) invariant to changes in the scale of the map for map points that are far apart. It also means that large clusters of points that are far apart interact in just the same way as individual points, so the optimization operates in the same way at all but the finest scales. A theoretical justification for our
>
> selection of the Student t-distribution is that it is closely related to the Gaussian distribution, as the Student t-distribution is an infinite mixture of Gaussians. A computationally convenient property is that it is much faster to evaluate the density of a point under a Student t-distribution than under a Gaussian because it does not involve an exponential, even though the Student t-distribution is equivalent to an infinite mixture of Gaussians with different variances.

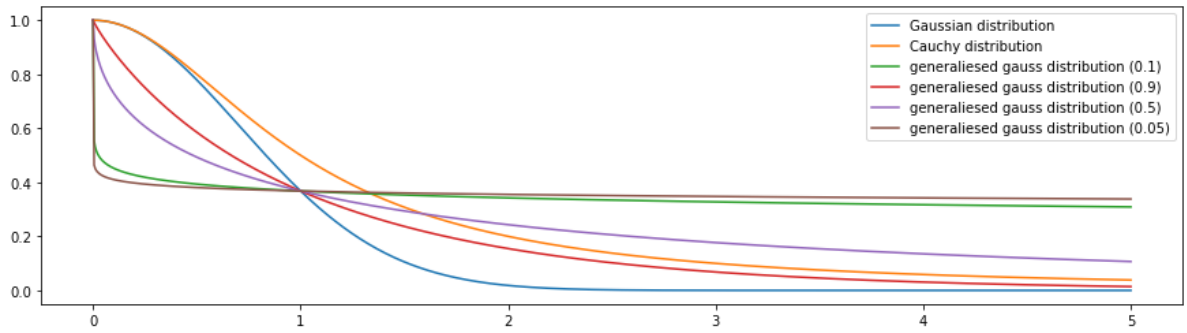
Замена в TSNE из sklearn.manifold

Ввод [20]:

```
# dist = pdist(X_embedded, "sqeuclidean")
# dist /= degrees_of_freedom
# dist += 1.0
# dist **= (degrees_of_freedom + 1.0) / -2.0
# Q = np.maximum(dist / (2.0 * np.sum(dist)), MACHINE_EPSILON)


# beta = 0.2
# dist = pdist(X_embedded, "euclidean") ** beta
# dist = np.exp(-dist)
# Q = np.maximum(dist / np.sum(dist), MACHINE_EPSILON)
```

Ввод [18]:

```
z = np.linspace(0., 5., 1000)
gauss = np.exp(-z**2)
cauchy = 1/(1+z**2)
g_01 = np.exp(-z**0.1)
g_05 = np.exp(-z**0.5)
g_09 = np.exp(-z**0.9)
g_005 = np.exp(-z**0.05)

plt.figure(figsize=(15, 4))
plt.plot(z, gauss, label='Gaussian distribution')
plt.plot(z, cauchy, label='Cauchy distribution')
plt.plot(z, g_01, label='generaliesed gauss distribution (0.1)')
plt.plot(z, g_09, label='generaliesed gauss distribution (0.9)')
plt.plot(z, g_05, label='generaliesed gauss distribution (0.5)')
plt.plot(z, g_005, label='generaliesed gauss distribution (0.05)')
plt.legend()
plt.show()
```



IRIS

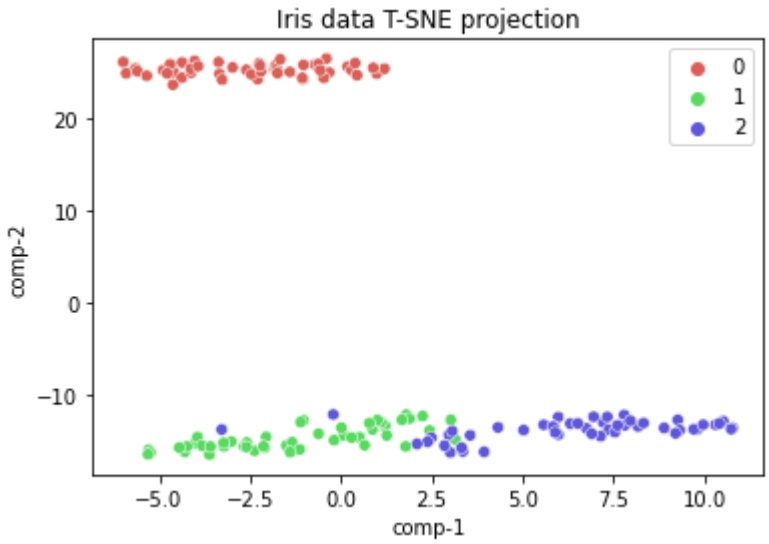
Ввод [9]:

```
iris = load_iris()
x = iris.data
y = iris.target
```

Ввод [27]:

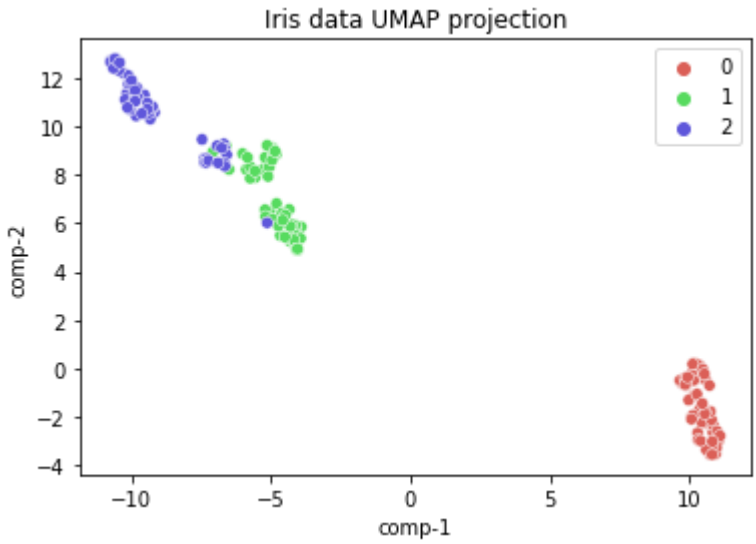
```
tsne = TSNE(n_components=2, random_state=123, method='exact')
z = tsne.fit_transform(x)

df = pd.DataFrame()
df["y"] = y
df["comp-1"] = z[:,0]
df["comp-2"] = z[:,1]

sns.scatterplot(x="comp-1", y="comp-2", hue=df.y.tolist(),
                palette=sns.color_palette("hls", 3),
                data=df).set(title="Iris data T-SNE projection")
plt.show()
```

```
/opt/anaconda3/lib/python3.8/site-packages/sklearn/manifold/_t_sne.py:
780: FutureWarning: The default initialization in TSNE will change fro
m 'random' to 'pca' in 1.2.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/sklearn/manifold/_t_sne.py:
790: FutureWarning: The default learning rate in TSNE will change from
200.0 to 'auto' in 1.2.
  warnings.warn(
```

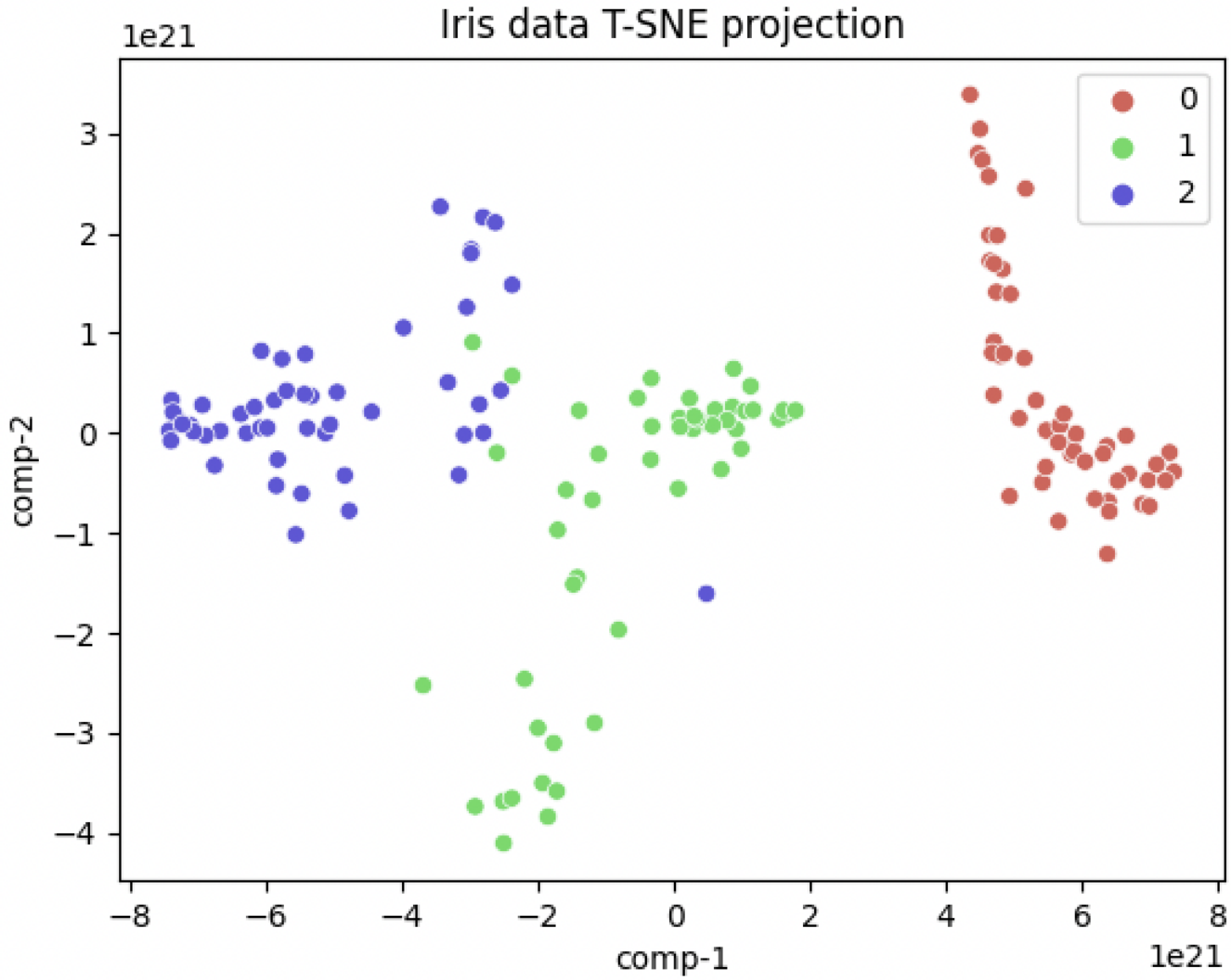


Ввод [28]:

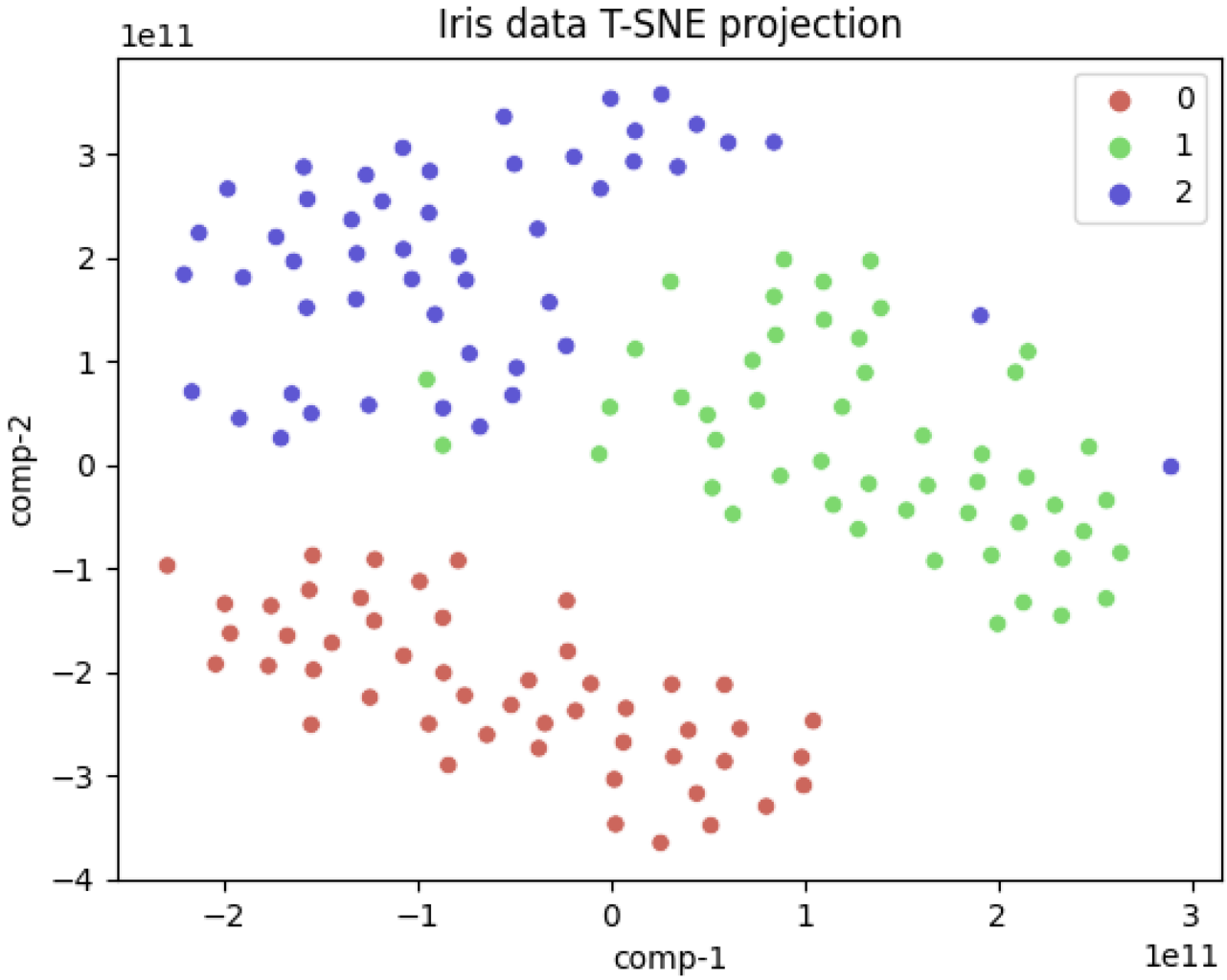
```
my_umap = umap.UMAP(n_components=2, random_state=123)
z = my_umap.fit_transform(x)

df = pd.DataFrame()
df["y"] = y
df["comp-1"] = z[:,0]
df["comp-2"] = z[:,1]

sns.scatterplot(x="comp-1", y="comp-2", hue=df.y.tolist(),
                palette=sns.color_palette("hls", 3),
                data=df).set(title="Iris data UMAP projection")
plt.show()
```



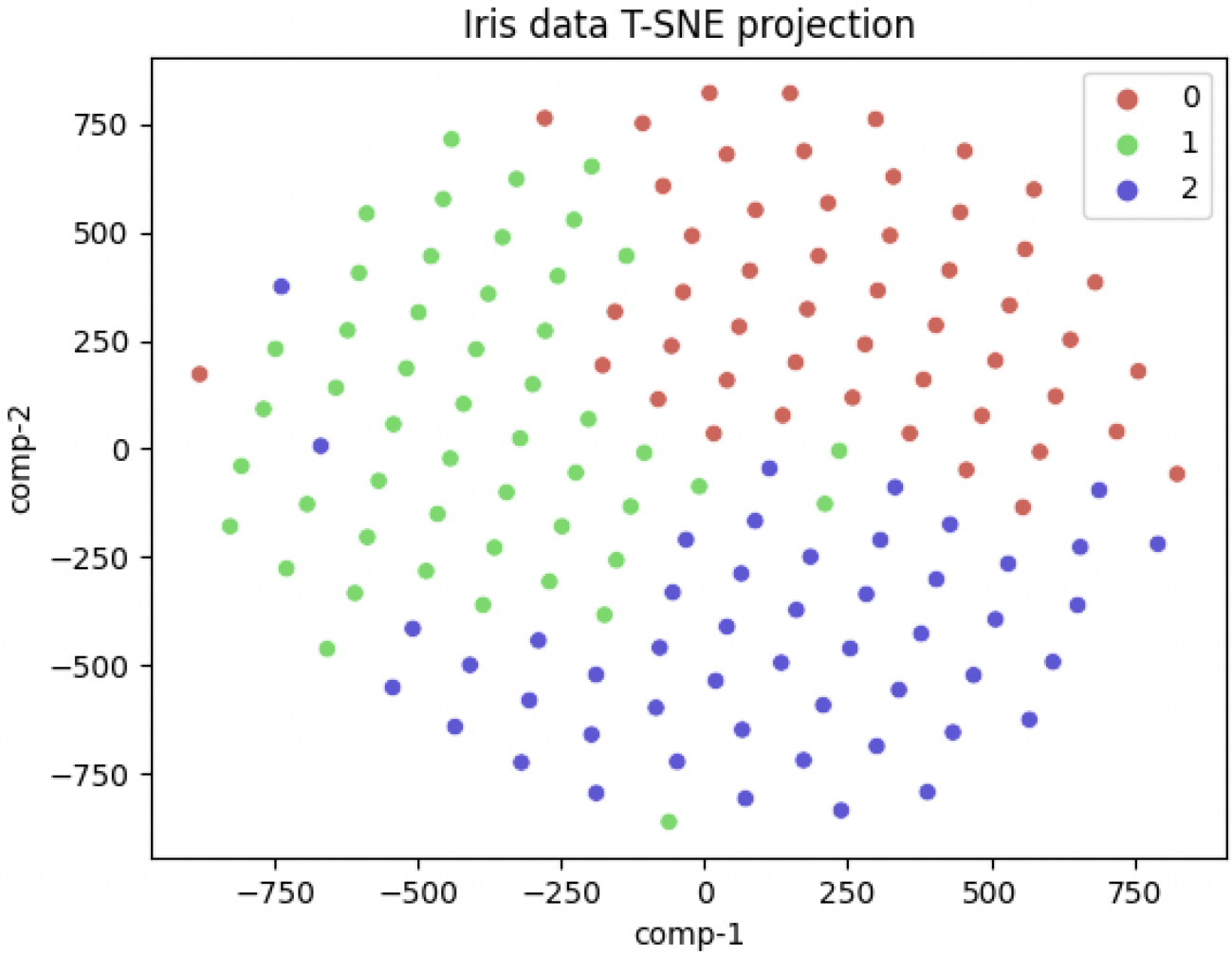
GSNE b=0.05


Iris data T-SNE projection
1e21
3
2
1
0
−1
−2
−3
−4
comp-2
−8
−6
−4
−2
0
2
4
6
8
comp-1
1e21
0
1
2

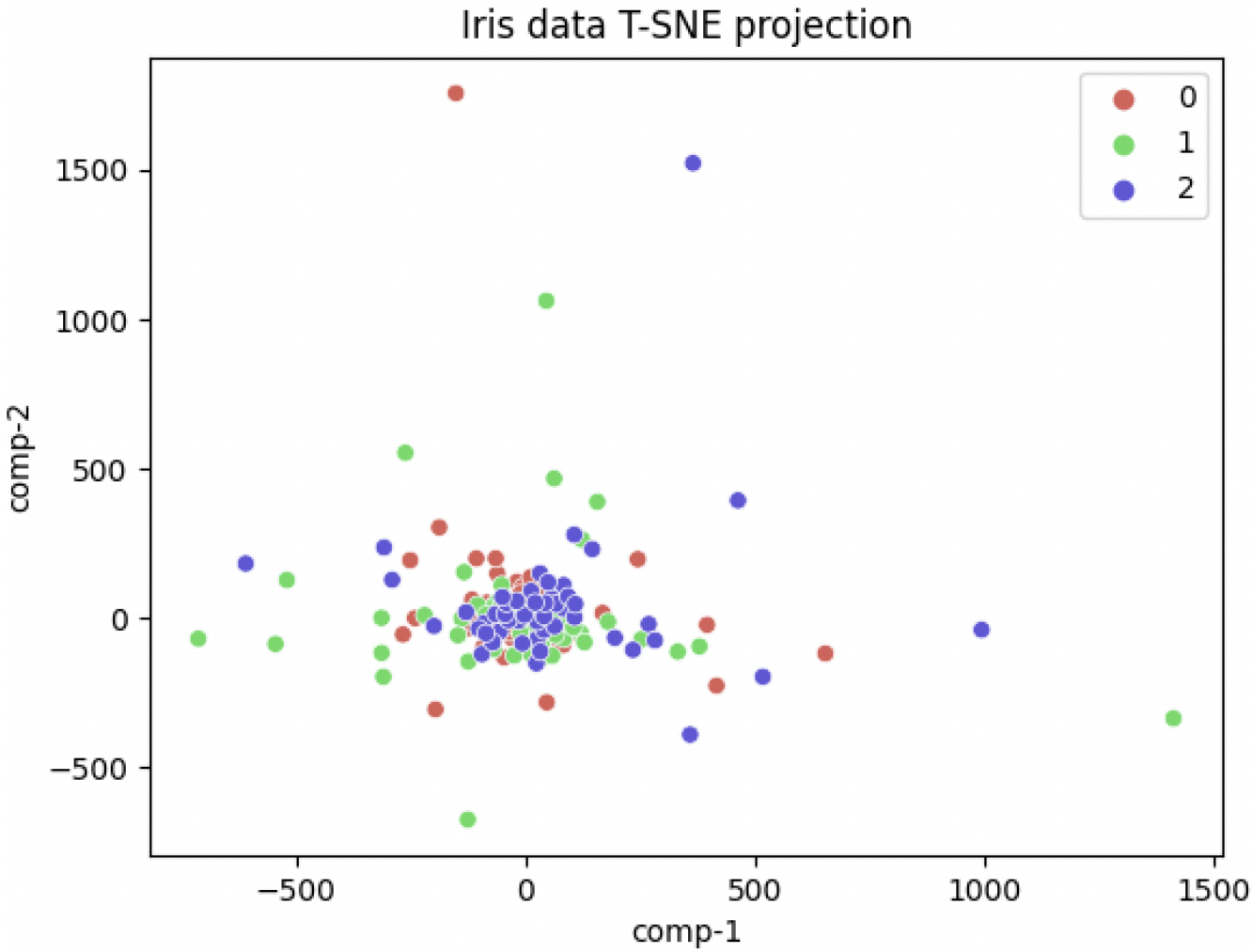
GSNE b=0.1


Iris data T-SNE projection
1e11
comp-2
comp-1
1e11
0
1
2
3
2
1
0
−1
−2
−3
−4
−2
−1
0
1
2
3

GSNE b=0.5


Iris data T-SNE projection
0
1
2
750
500
250
0
−250
−500
−750
comp-2
−750
−500
−250
0
250
500
750
comp-1

GSNE b=0.9



DIGITS

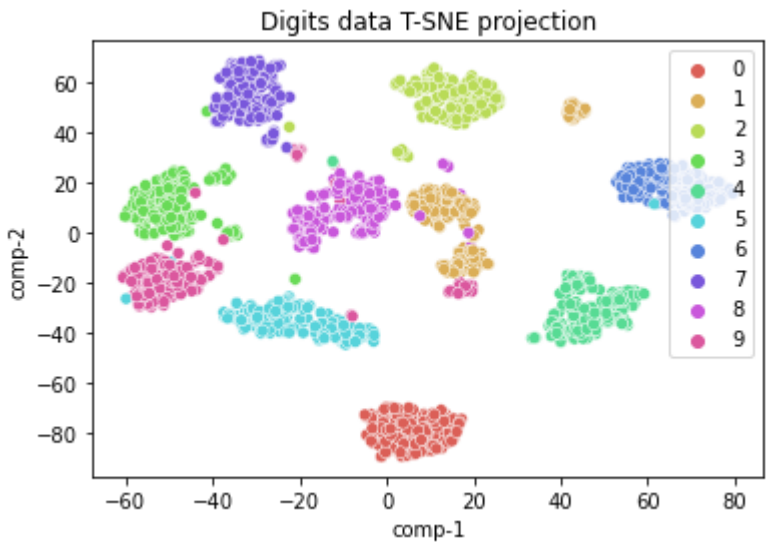
Ввод [29]:

```
from sklearn.datasets import load_digits

digits = load_digits()

x, y = digits['data'], digits['target']
```

Ввод [19]:

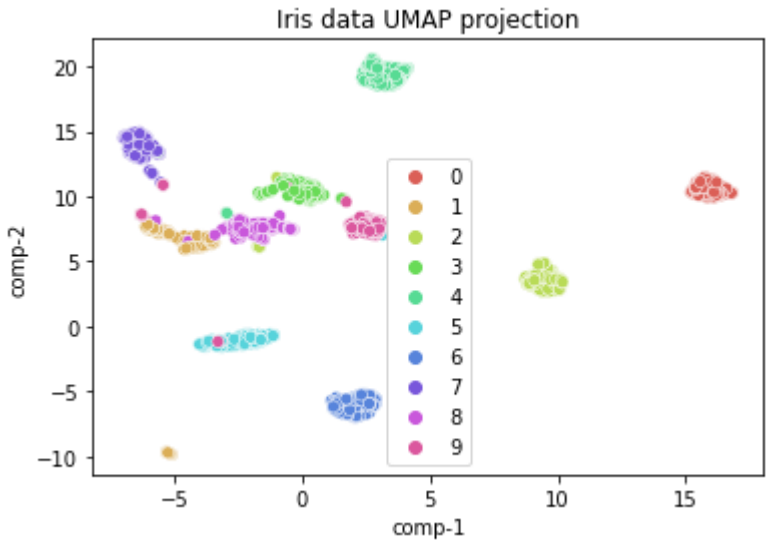
```
tsne = TSNE(n_components=2, random_state=123, method='exact')
z = tsne.fit_transform(x)

df = pd.DataFrame()
df["y"] = y
df["comp-1"] = z[:,0]
df["comp-2"] = z[:,1]

sns.scatterplot(x="comp-1", y="comp-2", hue=df.y.tolist(),
                palette=sns.color_palette("hls", 10),
                data=df).set(title="Digits data T-SNE projection")
plt.show()
```

```
/opt/anaconda3/lib/python3.8/site-packages/sklearn/manifold/_t_sne.py:
780: FutureWarning: The default initialization in TSNE will change fro
m 'random' to 'pca' in 1.2.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/sklearn/manifold/_t_sne.py:
790: FutureWarning: The default learning rate in TSNE will change from
200.0 to 'auto' in 1.2.
  warnings.warn(
```

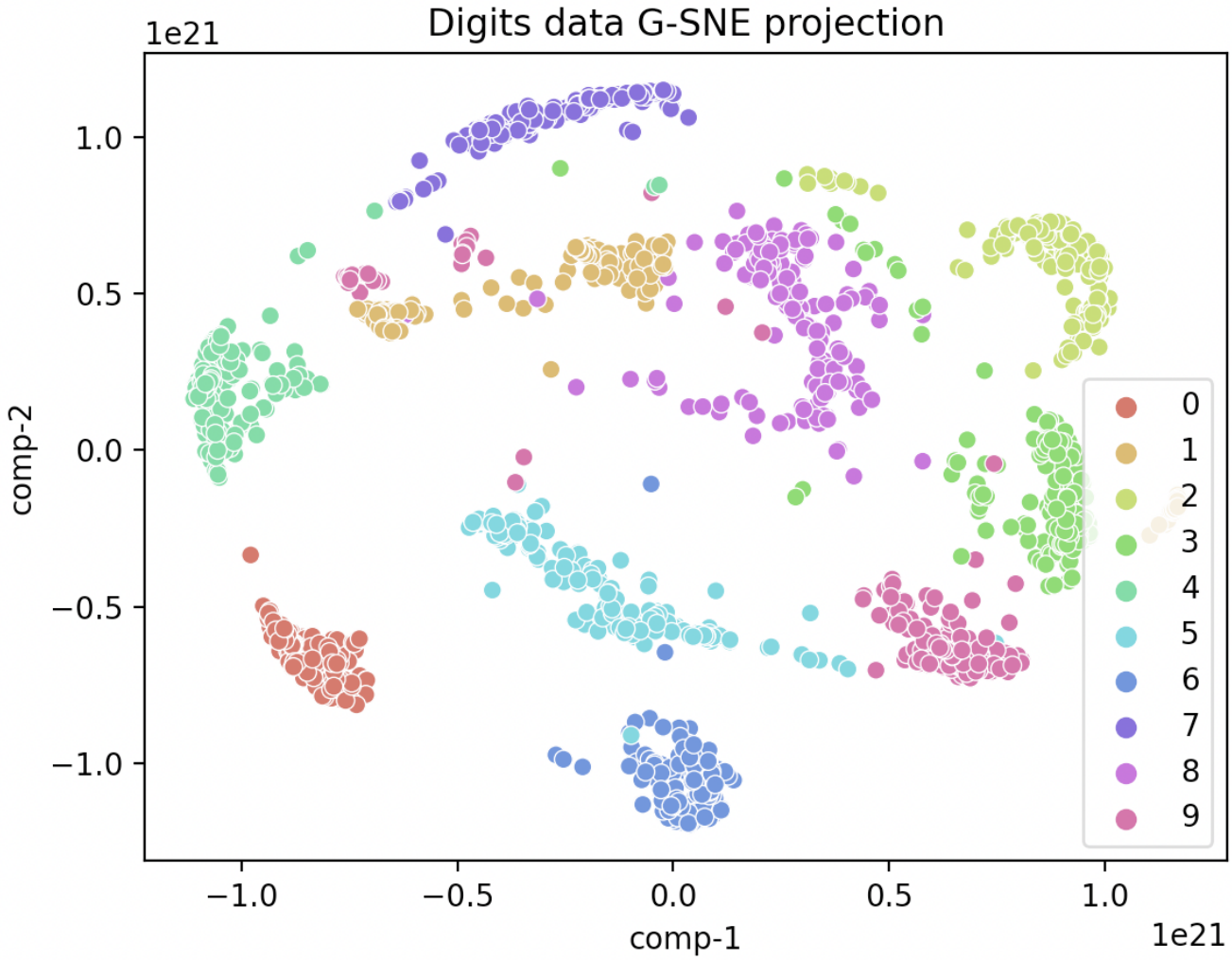


Ввод [24]:

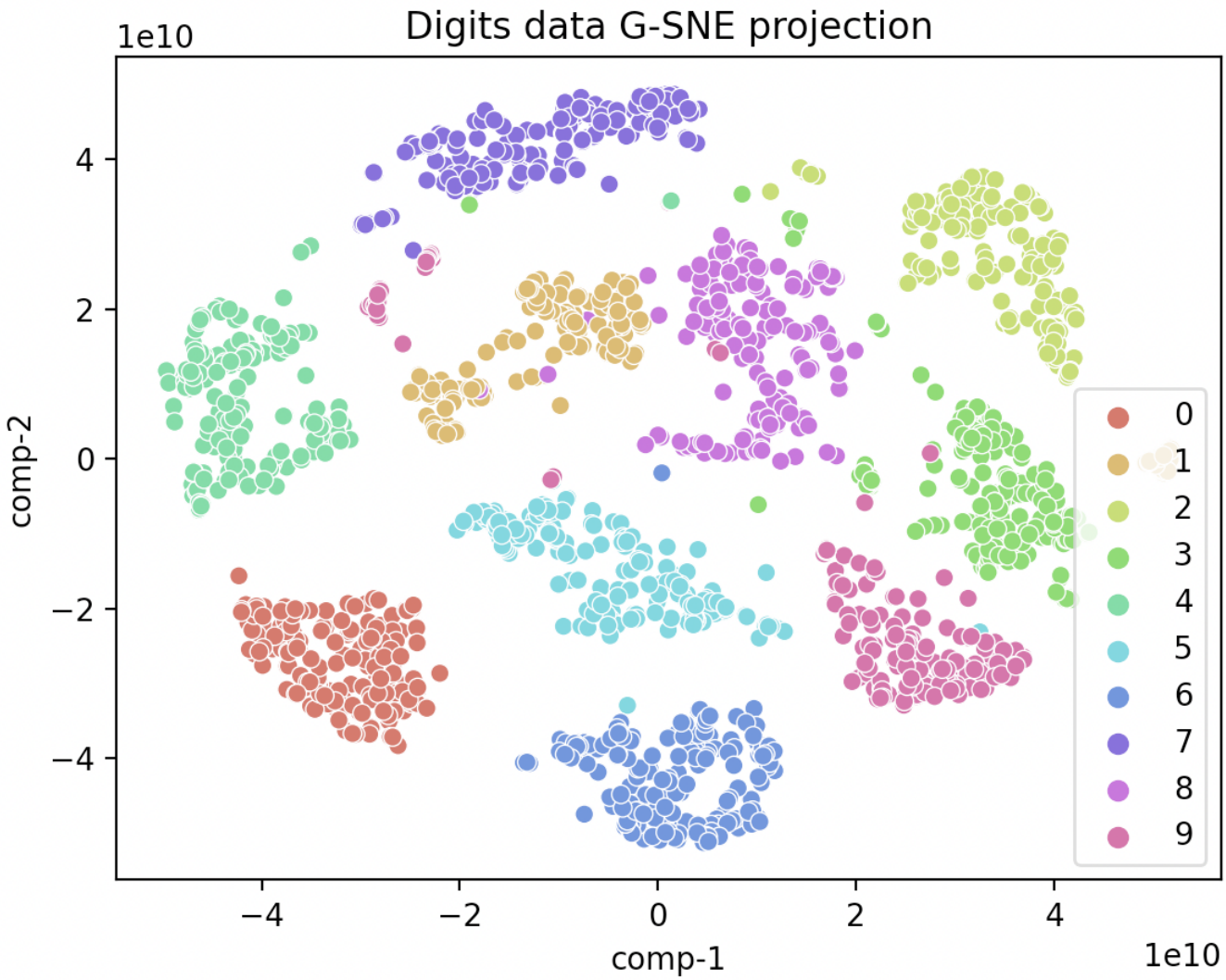
```
my_umap = umap.UMAP(n_components=2, random_state=123)
z = my_umap.fit_transform(x)

df = pd.DataFrame()
df["y"] = y
df["comp-1"] = z[:,0]
df["comp-2"] = z[:,1]

sns.scatterplot(x="comp-1", y="comp-2", hue=df.y.tolist(),
                palette=sns.color_palette("hls", 10),
                data=df).set(title="Digits data UMAP projection")
plt.show()
```



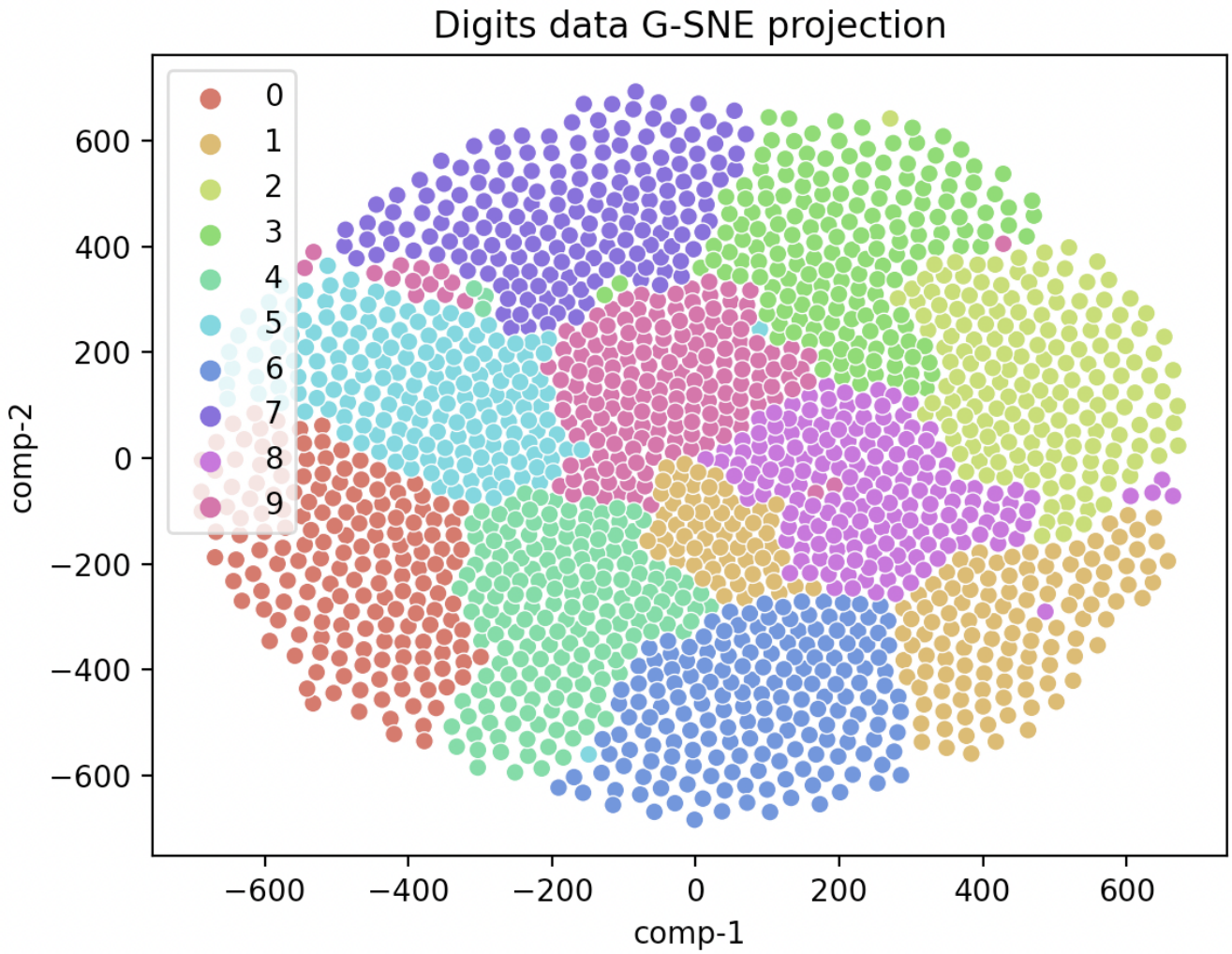
GSNE b=0.05


Digits data G-SNE projection
1e21
1.0
0.5
0.0
−0.5
−1.0
comp-2
−1.0
−0.5
0.0
0.5
1.0
comp-1
1e21
0
1
2
3
4
5
6
7
8
9

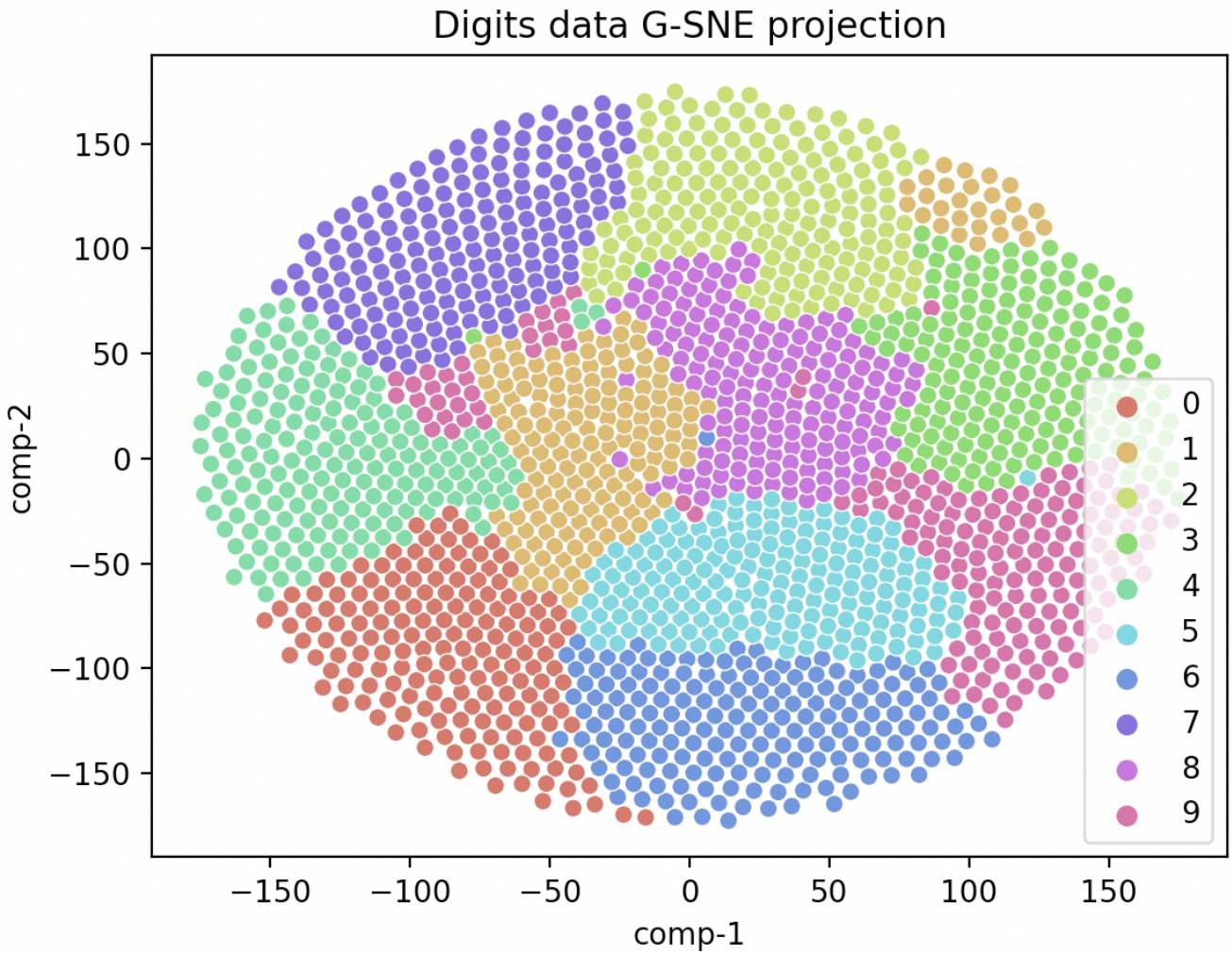
GSNE b=0.1


Digits data G-SNE projection
1e10
4
2
0
−2
−4
comp-2
−4
−2
0
2
4
comp-1
1e10
0
1
2
3
4
5
6
7
8
9

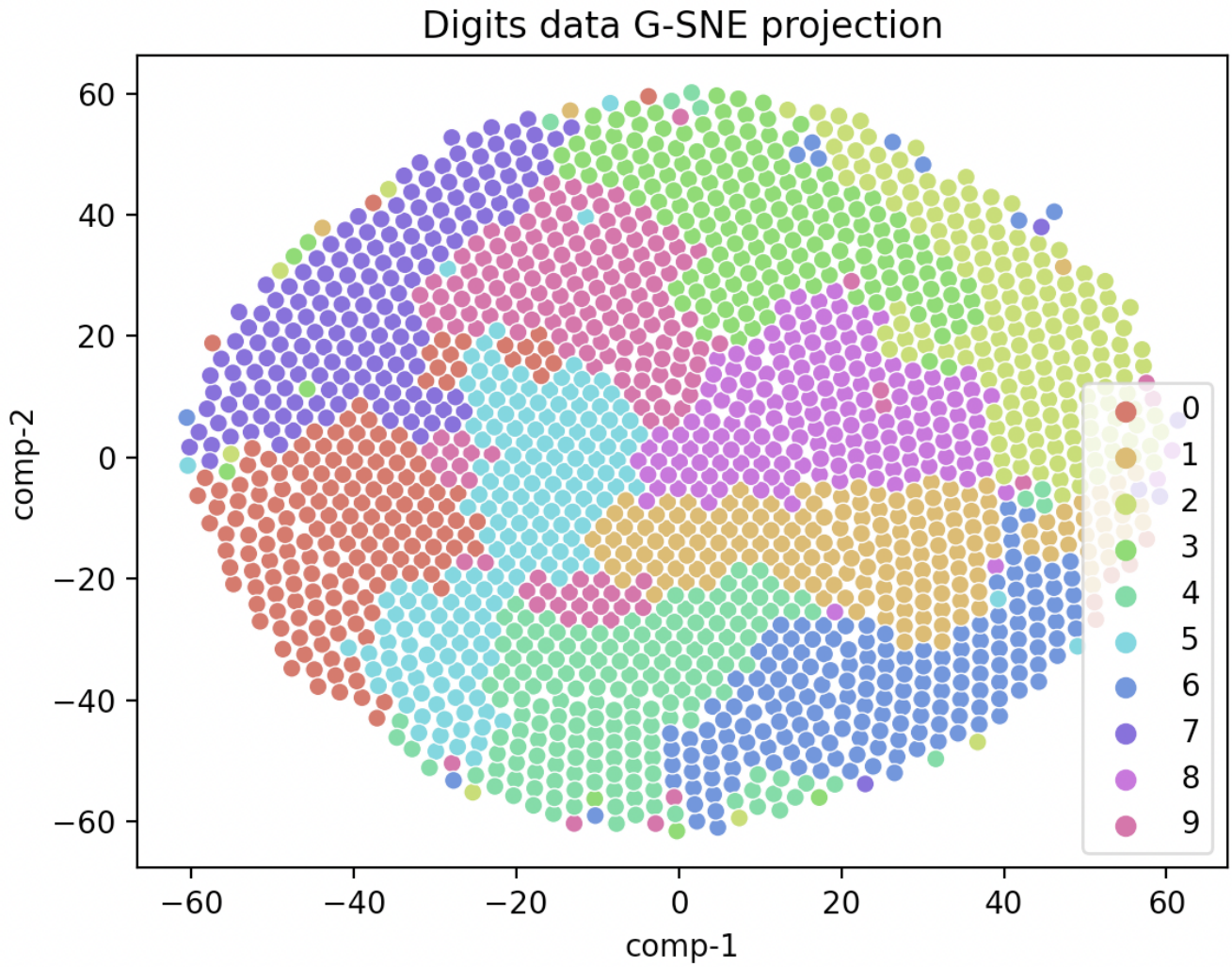
GSNE b=0.5


Digits data G-SNE projection
0
1
2
3
4
5
6
7
8
9
600
400
200
0
−200
−400
−600
comp-2
−600
−400
−200
0
200
400
600
comp-1

GSNE b=0.9


Digits data G-SNE projection
comp-2
comp-1
150
100
50
0
−50
−100
−150
−150
−100
−50
0
50
100
150
0
1
2
3
4
5
6
7
8
9

GSNE b=2



FASHION MNIST

Ввод [10]:

```
data = pd.read_csv('/Users/maxim_kocanov/Desktop/Учеба/Магистратура/науч/fashion-mnist_
data.head()
```

Out[10]:

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel77
0	2	0	0	0	0	0	0	0	0	0	...	0	
1	9	0	0	0	0	0	0	0	0	0	...	0	
2	6	0	0	0	0	0	0	0	5	0	...	0	
3	0	0	0	0	1	2	0	0	0	0	...	3	
4	3	0	0	0	0	0	0	0	0	0	...	0	

5 rows × 785 columns

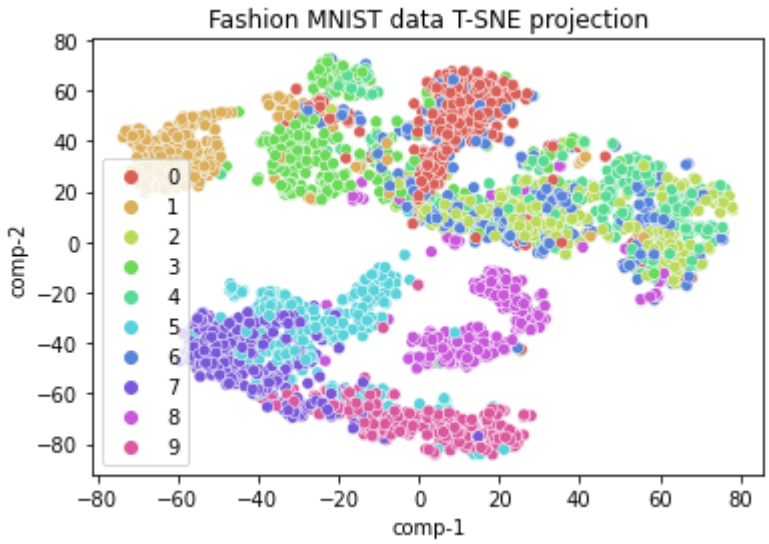
Ввод [11]:

```
y = np.array(data['label'])[:3000]
x = np.array(data.drop(['label'], axis=1))[:3000]
```

Ввод [12]:

```
tsne = TSNE(n_components=2, random_state=123, method='exact')
z = tsne.fit_transform(x)

df = pd.DataFrame()
df["y"] = y
df["comp-1"] = z[:,0]
df["comp-2"] = z[:,1]

sns.scatterplot(x="comp-1", y="comp-2", hue=df.y.tolist(),
                palette=sns.color_palette("hls", 10),
                data=df).set(title="Fashion MNIST data T-SNE projection")
plt.show()
```

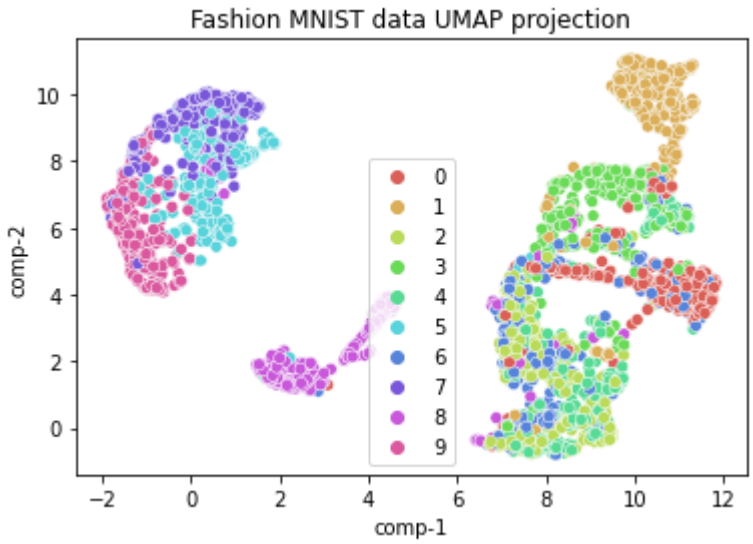
```
/opt/anaconda3/lib/python3.8/site-packages/sklearn/manifold/_t_sne.py:
780: FutureWarning: The default initialization in TSNE will change fro
m 'random' to 'pca' in 1.2.
  warnings.warn(
/opt/anaconda3/lib/python3.8/site-packages/sklearn/manifold/_t_sne.py:
790: FutureWarning: The default learning rate in TSNE will change from
200.0 to 'auto' in 1.2.
  warnings.warn(
```



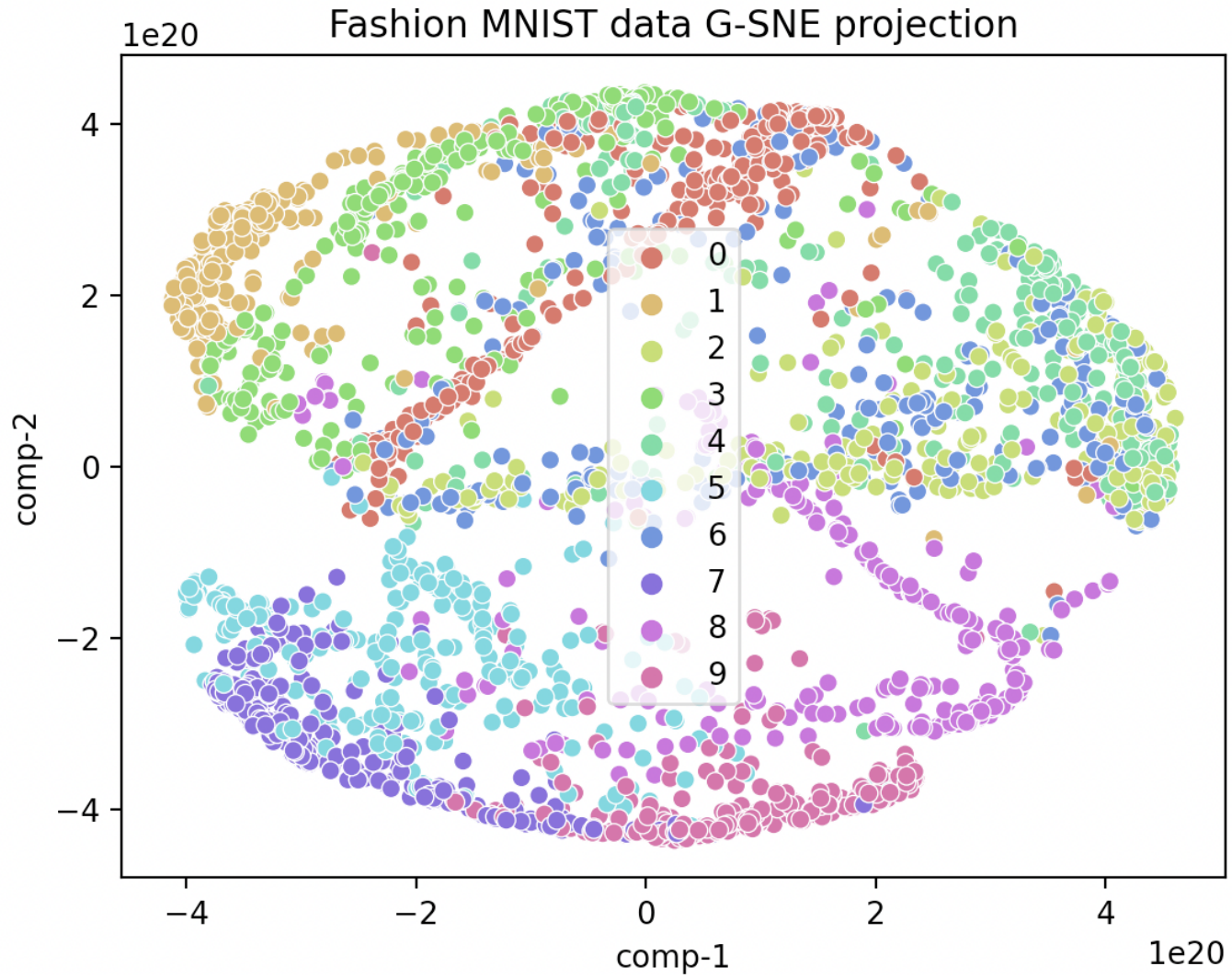
Ввод [13]:

```
my_umap = umap.UMAP(n_components=2, random_state=123)
z = my_umap.fit_transform(x)

df = pd.DataFrame()
df["y"] = y
df["comp-1"] = z[:,0]
df["comp-2"] = z[:,1]

sns.scatterplot(x="comp-1", y="comp-2", hue=df.y.tolist(),
                palette=sns.color_palette("hls", 10),
                data=df).set(title="Fashion MNIST data UMAP projection")
plt.show()
```

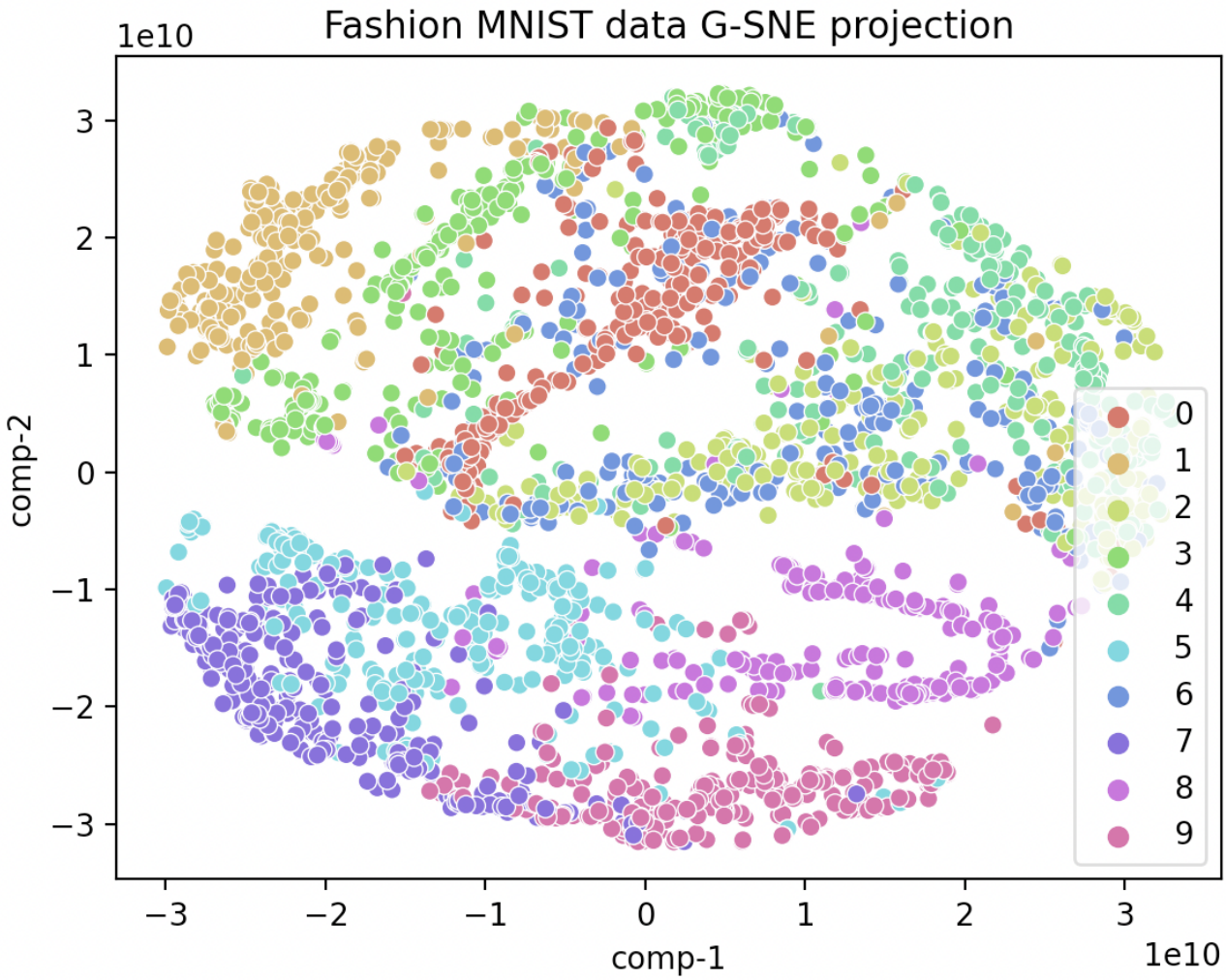
```
OMP: Info #271: omp_set_nested routine deprecated, please use omp_set_
max_active_levels instead.
```



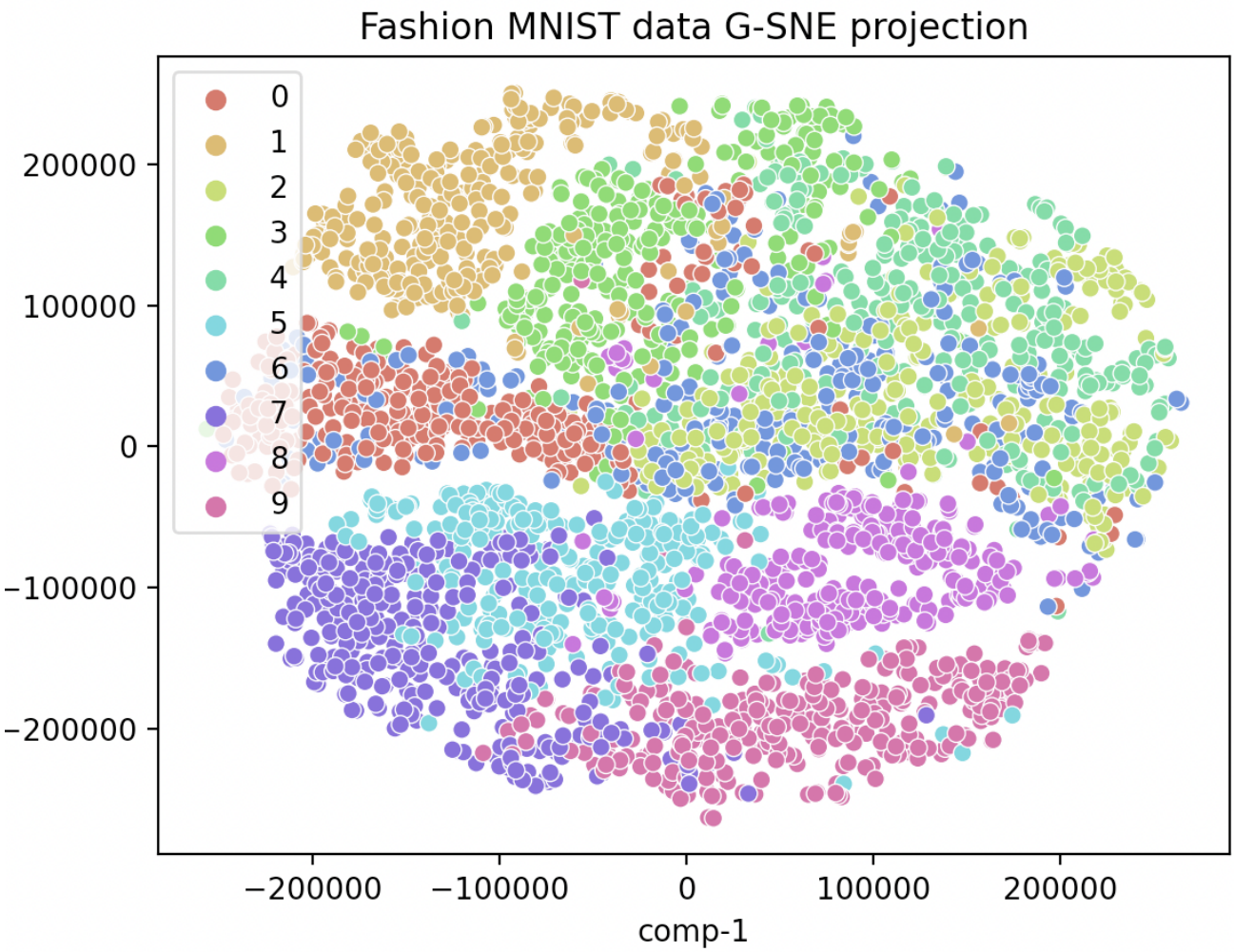
GSNE b=0.05


Fashion MNIST data G-SNE projection
1e20
4
2
0
−2
−4
comp-2
−4
−2
0
2
4
comp-1
1e20
0
1
2
3
4
5
6
7
8
9

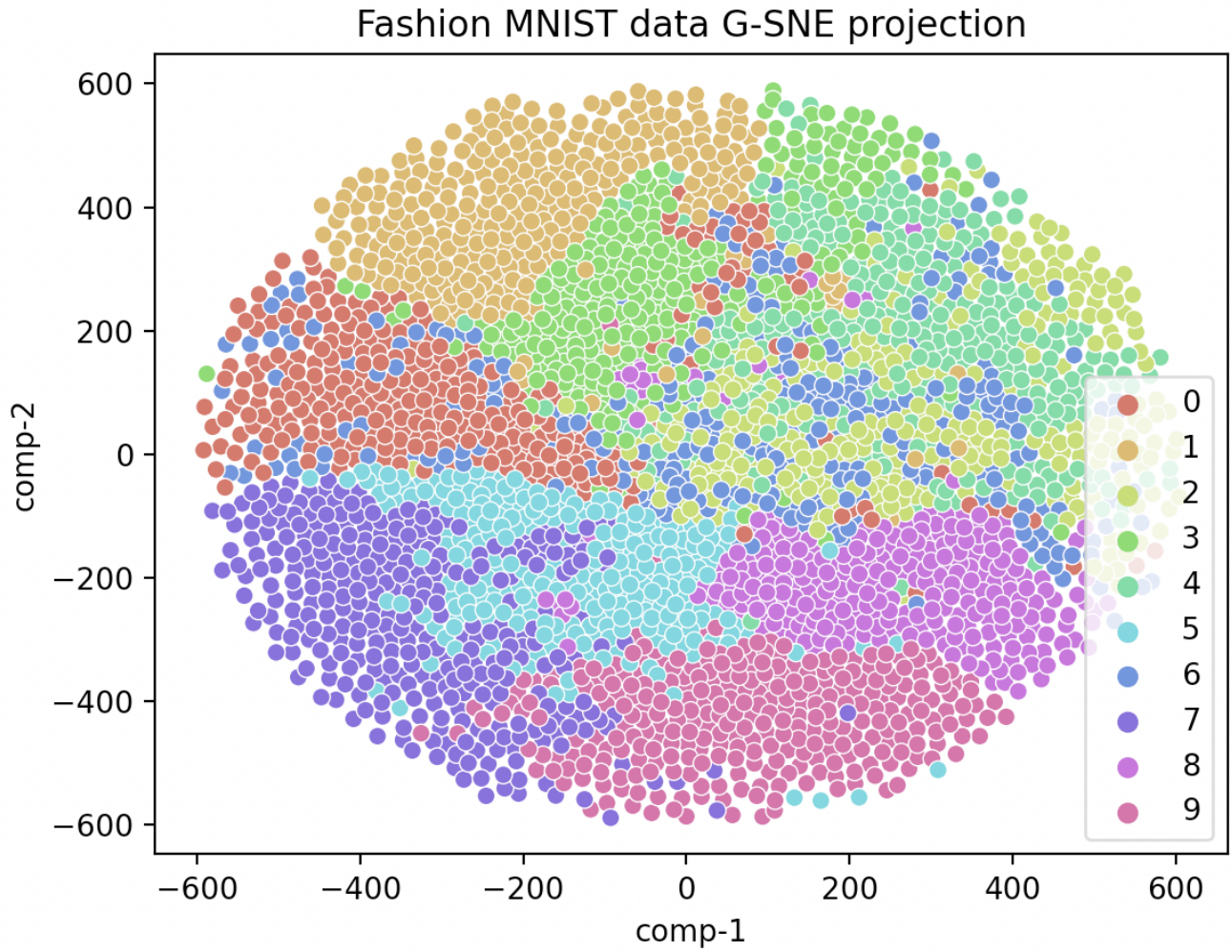
GSNE b=0.1


Fashion MNIST data G-SNE projection
1e10
3
2
1
0
−1
−2
−3
comp-2
−3
−2
−1
0
1
2
3
comp-1
1e10
0
1
2
3
4
5
6
7
8
9

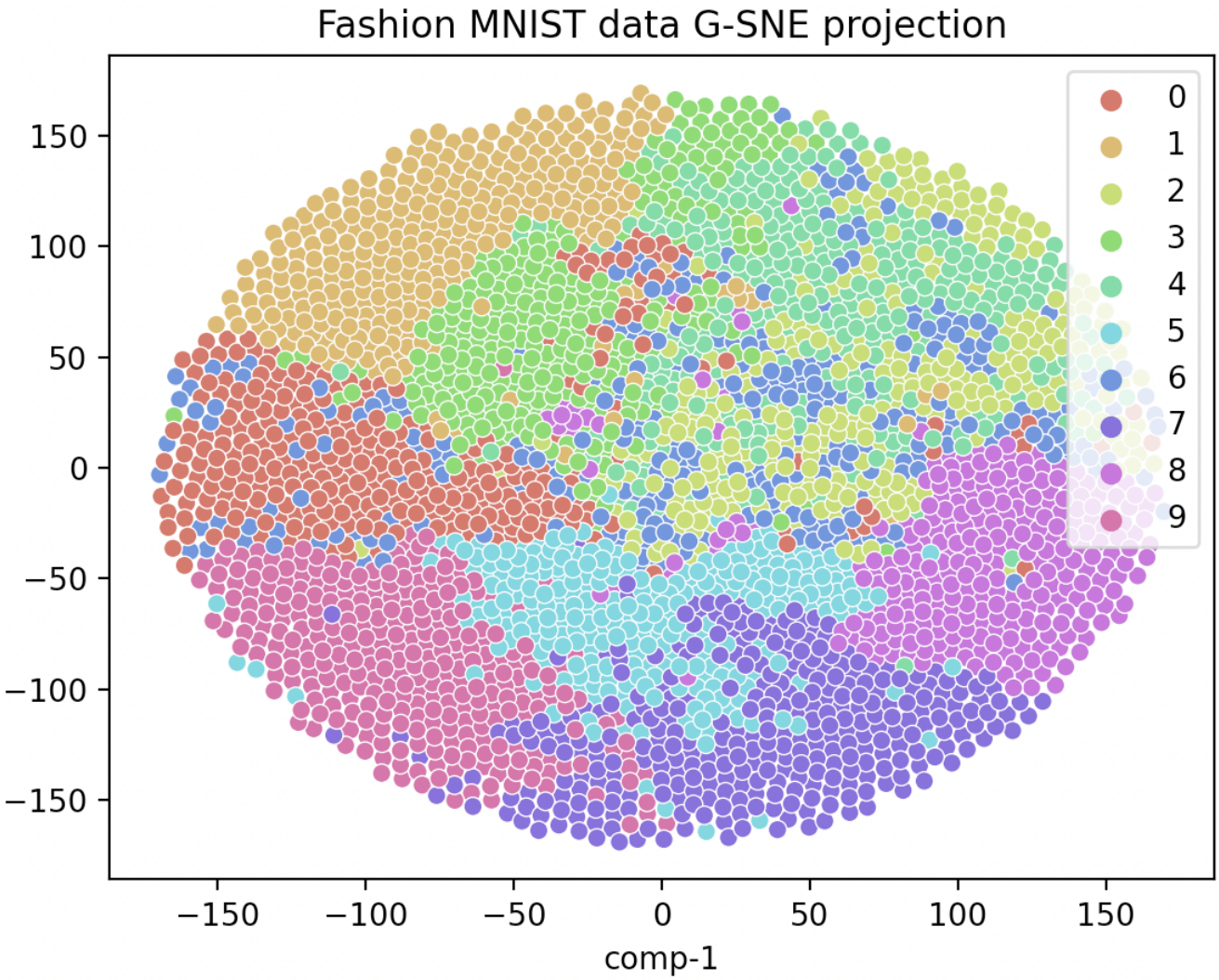
GSNE b=0.2


Fashion MNIST data G-SNE projection
0
1
2
3
4
5
6
7
8
9
200000
100000
0
−100000
−200000
−200000
−100000
0
100000
200000
comp-1

GSNE b=0.5


Fashion MNIST data G-SNE projection
comp-2
comp-1
600
400
200
0
−200
−400
−600
−600
−400
−200
0
200
400
600
0
1
2
3
4
5
6
7
8
9

GSNE b=0.9


Fashion MNIST data G-SNE projection
150
100
50
0
−50
−100
−150
−150
−100
−50
0
50
100
150
comp-1
0
1
2
3
4
5
6
7
8
9

Ввод [ ]: