Как сделать так, чтобы сегменты между маркерами рисовались в пространстве трех главных компонент метода PCA?

1. *"""Utility code to visualize a minimum spanning tree"""*X = model.X\_fit\_  
   fig**,** ax = plt.subplots(**1, 2,** figsize=(**16, 6**)**,** sharex=True**,** sharey=True)  
   for axi**,** full\_graph**,** colors in zip(ax**,** [True**,** False]**,** ['lightblue'**,** model.labels\_]):  
    segments = model.get\_graph\_segments(full\_graph=full\_graph)  
    print(segments)  
    axi.plot(segments[**0**]**,** segments[**1**]**,** '-k'**,** zorder=**1,** lw=**1**)  
    axi.scatter(X[:**, 0**]**,** X[:**, 1**]**,** c=colors**,** cmap=cmap**,** zorder=**2**)  
    axi.axis('tight')  
      
     
   ax[**0**].set\_title('Full Minimum Spanning Tree'**,** size=**16**)  
   ax[**1**].set\_title('Trimmed Minimum Spanning Tree'**,** size=**16**);  
   plt.show()

Чтобы отобразить сегменты между маркерами в пространстве трёх главных компонент с использованием метода PCA (Principal Component Analysis), нужно модифицировать код так, чтобы он сначала проецировал данные в пространство трёх главных компонент, а затем построил сегменты и маркеры в этом новом пространстве.

Вот как можно модифицировать вашу функцию:

1. Провести преобразование PCA для получения трёх главных компонент.
2. Использовать проекции этих компонент для отображения точек и сегментов.
3. Изменить визуализацию на 3D с использованием Axes3D для отображения сегментов в пространстве трёх компонент.

import numpy as np

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

from sklearn.decomposition import PCA

def plot\_mst\_3d(model, cmap='rainbow'):

"""Функция для визуализации минимального остовного дерева в пространстве трех главных компонент"""

# Получаем данные, на которых был натренирован модель

X = model.X\_fit\_

# Применяем PCA для проекции в пространство трех главных компонент

pca = PCA(n\_components=3)

X\_pca = pca.fit\_transform(X)

# Создаем 3D-график

fig = plt.figure(figsize=(16, 8))

ax1 = fig.add\_subplot(121, projection='3d')

ax2 = fig.add\_subplot(122, projection='3d')

for ax, full\_graph, colors in zip([ax1, ax2], [True, False], ['lightblue', model.labels\_]):

segments = model.get\_graph\_segments(full\_graph=full\_graph)

# Для каждого сегмента рисуем линию в пространстве PCA

for seg in segments:

p1 = X\_pca[seg[0]]

p2 = X\_pca[seg[1]]

ax.plot([p1[0], p2[0]], [p1[1], p2[1]], [p1[2], p2[2]], '-k', zorder=1, lw=1)

# Рисуем точки

scatter = ax.scatter(X\_pca[:, 0], X\_pca[:, 1], X\_pca[:, 2], c=colors, cmap=cmap, zorder=2)

ax.set\_title('Full MST' if full\_graph else 'Trimmed MST', size=16)

plt.show()

**Объяснение кода:**

1. **PCA**: Мы используем PCA(n\_components=3), чтобы проецировать данные на пространство первых трёх главных компонент.
2. **3D график**: Мы используем Axes3D для создания трёхмерного графика, чтобы можно было визуализировать данные в пространстве этих компонент.
3. **Рисование сегментов**: В цикле для каждого сегмента мы рисуем линии между точками, используя координаты из пространства PCA.

Эта функция будет отображать минимальное остовное дерево в пространстве трёх главных компонент, что позволяет более глубоко анализировать структуру данных.