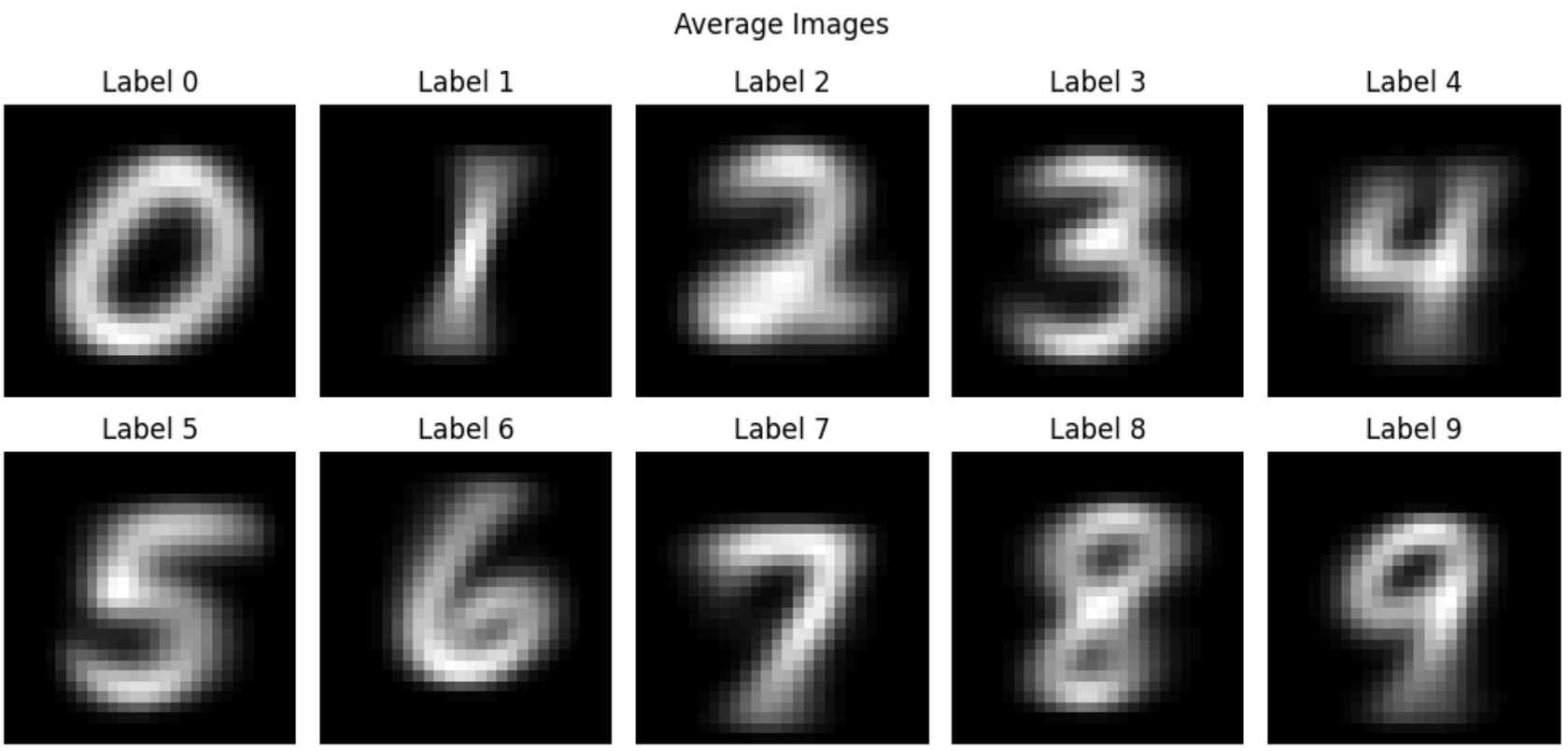
In [ ]:
```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from scipy.spatial.distance import euclidean
from scipy.spatial.distance import cosine
import networkx as nx
import matplotlib.pyplot as plt
#%matplotlib qt
```

Mnist veri setini import ediyorum. Amacım el yazısıyla yazılmış rakamları kullanarak rakamların birbiri arasındaki benzerliğin analizini yapmak.

In [ ]:
```
# MNIST veri setini yükle
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Veriyi normalize et
x_train = x_train / 255.0
x_test = x_test / 255.0
```

Her sayının ortalama görüntüsünü hesaplayıp görselleştiriyorum

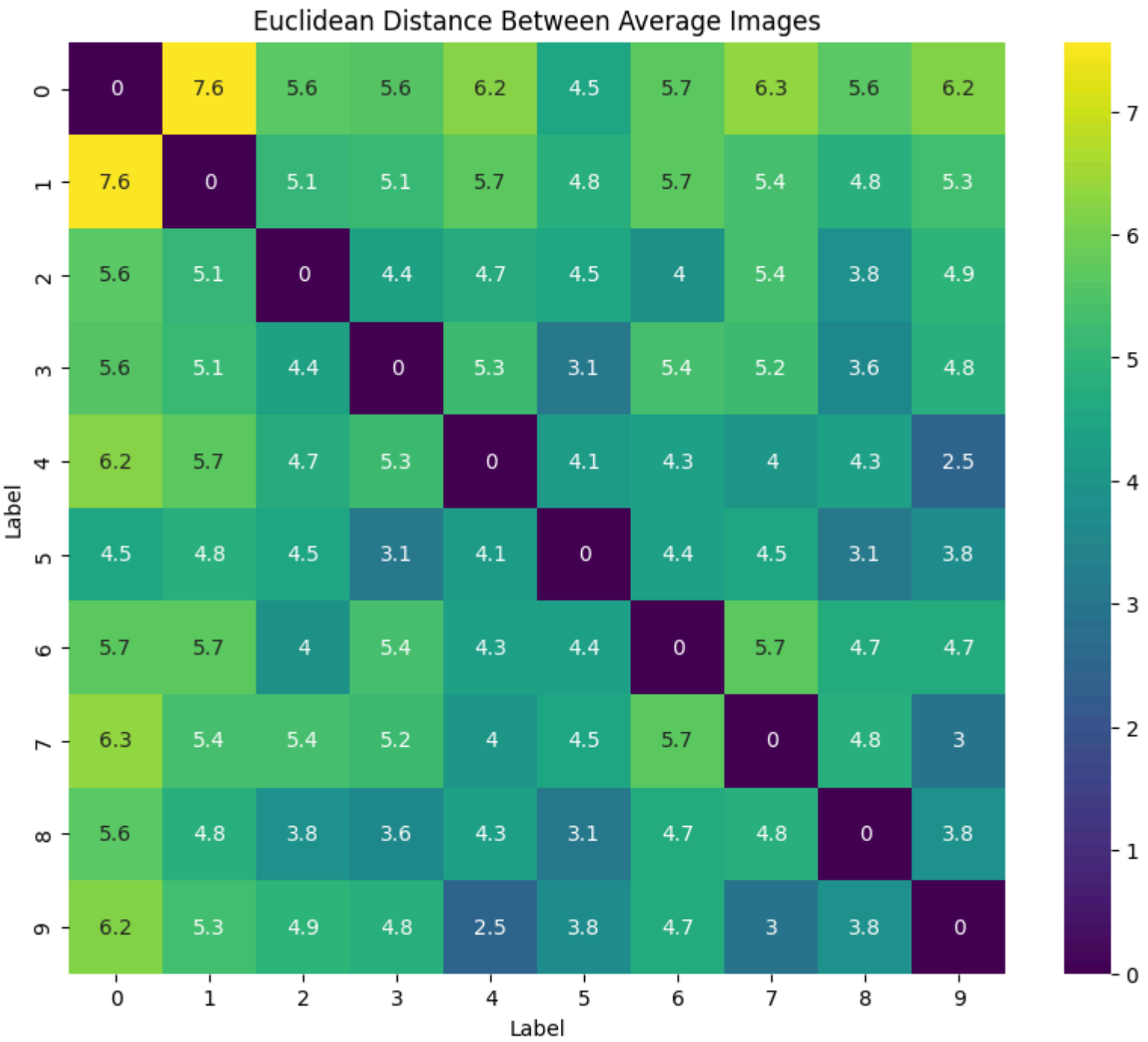
In [ ]:
```
# Her etiket için ortalama görüntü hesapla
average_images = np.zeros((10, 28, 28))

for i in range(10):
    indices = np.where(y_train == i)
    average_images[i] = np.mean(x_train[indices], axis=0)

# Ortalama görüntüleri görselleştir

fig, axes = plt.subplots(2, 5, figsize=(10, 5))
axes = axes.flatten()
for i in range(10):
    axes[i].imshow(average_images[i], cmap='gray')
    axes[i].set_title(f'Label {i}')
    axes[i].axis('off')

plt.suptitle('Average Images')
plt.tight_layout()
plt.show()
```

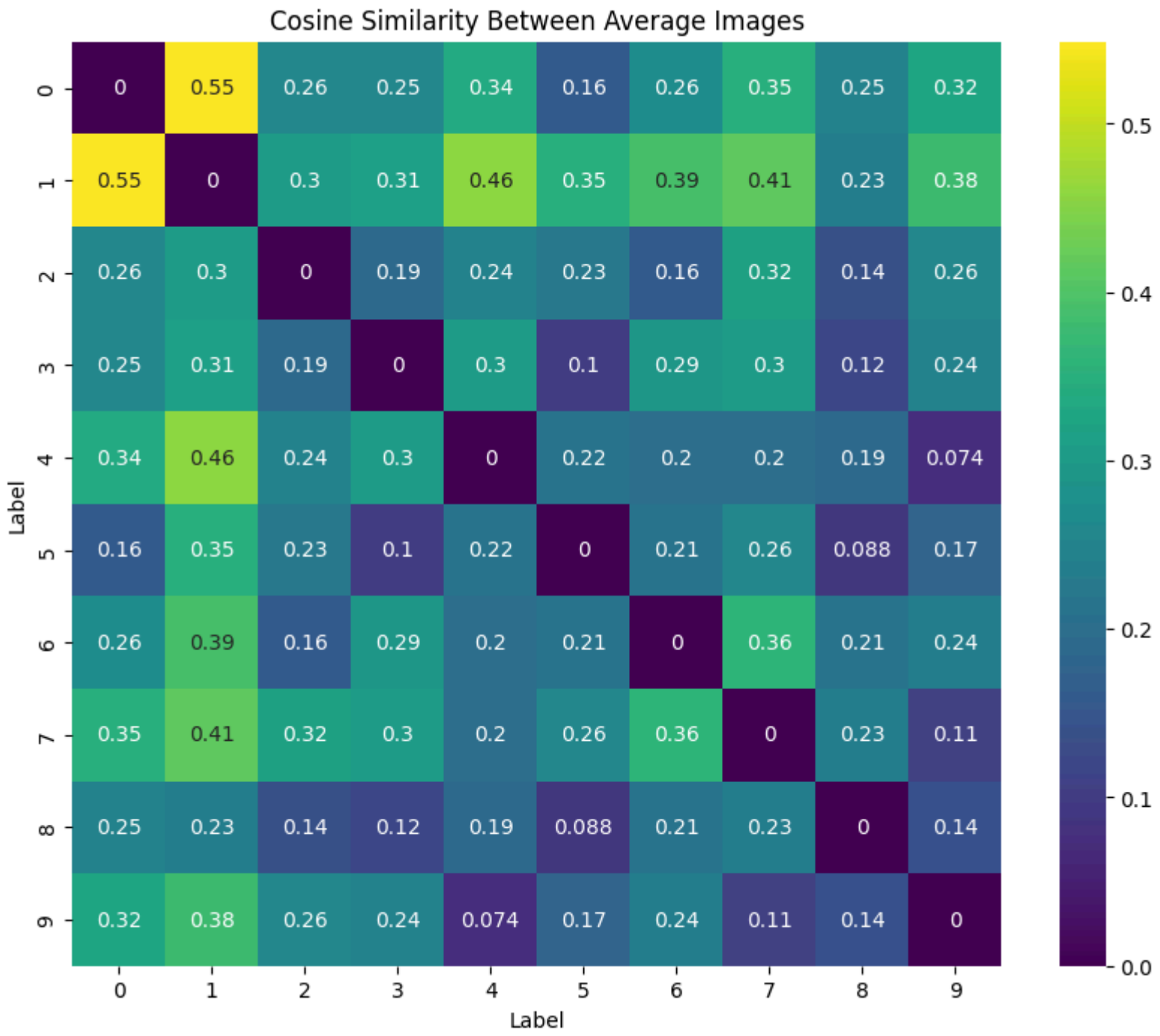


Öklid mesafesi ve Cosin mesafesini kullanark benzerlikleri hesaplıyorum. Benzerlik matrislerini plotluyorum

In [ ]:
```
# Öklidyen mesafesi ile benzerlik matrisi
euclidean_distances = np.zeros((10, 10))
cosine_similarities = np.zeros((10, 10))

for i in range(10):
    for j in range(10):
        euclidean_distances[i, j] = euclidean(average_images[i].flatten(), average_images[j].flatten())
        cosine_similarities[i, j] = cosine(average_images[i].flatten(), average_images[j].flatten())

# Sonuçları görselleştir
import seaborn as sns

plt.figure(figsize=(10, 8))
sns.heatmap(euclidean_distances, annot=True, cmap='viridis')
plt.title('Euclidean Distance Between Average Images')
plt.xlabel('Label')
plt.ylabel('Label')
```

```
plt.show()

plt.figure(figsize=(10, 8))
sns.heatmap(cosine_similarities, annot=True, cmap='viridis')
plt.title('Cosine Similarity Between Average Images')
plt.xlabel('Label')
plt.ylabel('Label')
plt.show()
print("euclidean_distance ortalama: ",np.mean(euclidean_distances))
print("cosine_similarity ortalama: ",np.mean(cosine_similarities))
```



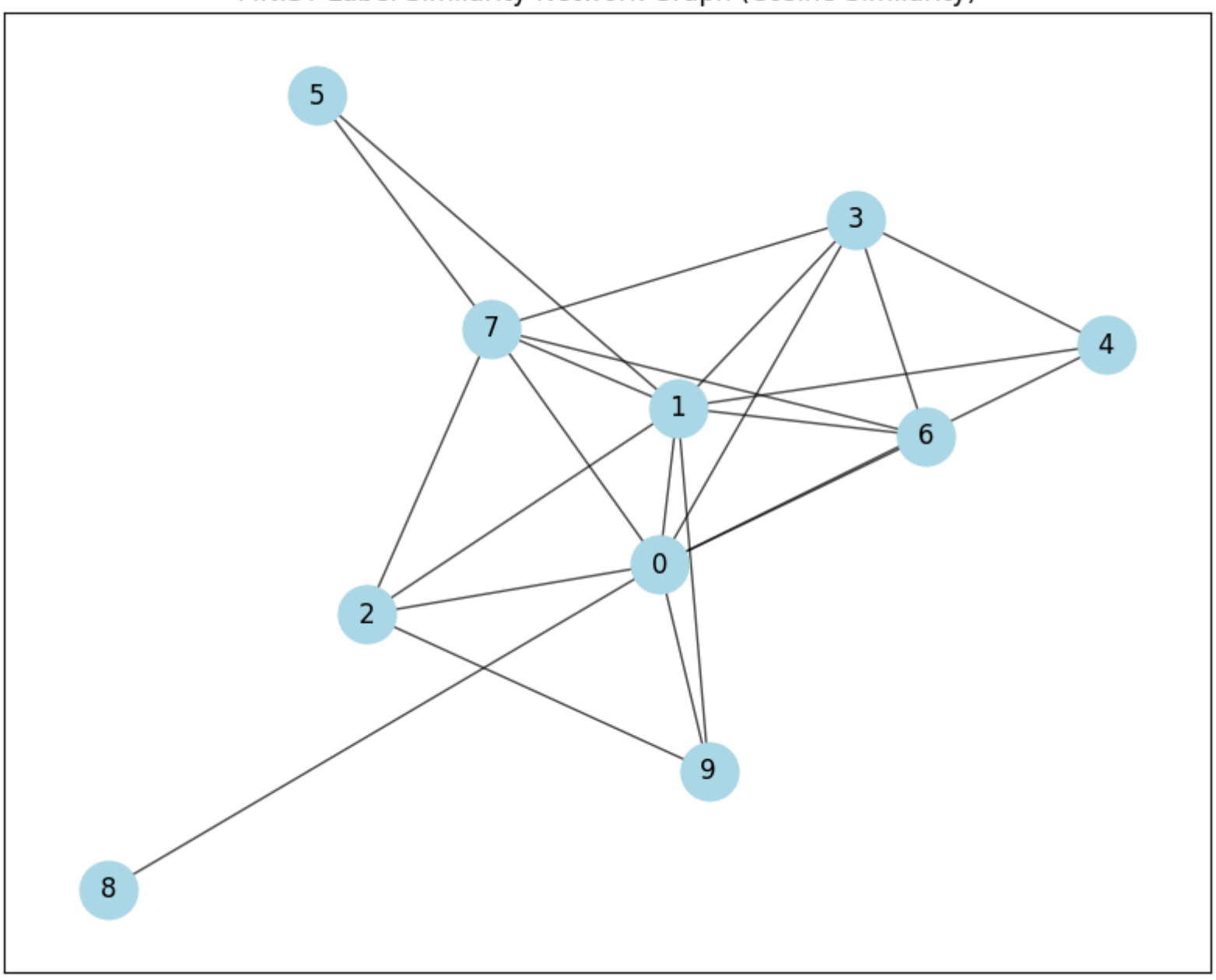



```
euclidean_distance ortalama:  4.319241191814074
cosine_similarity ortalama:  0.22755095169802161
```

Matrislerin ortalamasını göz önünde bulundurarak bir treshold belirliyorum ve bu eşik değeri geçerlerse grafiğe edge olarak ekliyorum. Benzerlik değerleri birbirine yakınlık ve uzaklıklarını belirlyo.

In [ ]:

```python
# NetworkX graph oluştur
G = nx.Graph()

# Düğümleri ekle
for i in range(10):
    G.add_node(i, label=str(i))

# Kenarları ekle (benzerlik eşiği belirleyerek)
threshold = 0.25  # Benzerlik eşiği, bunu ayarlayabilirsiniz
for i in range(10):
    for j in range(i + 1, 10):
        if cosine_similarities[i, j] >= threshold:
            G.add_edge(i, j, weight=cosine_similarities[i, j])

# Graph'ı çiz
plt.figure(figsize=(10, 8))
pos = nx.spring_layout(G, seed=42)  # Düğüm yerleşimini belirle
nx.draw_networkx_nodes(G, pos, node_size=700, node_color='lightblue')
nx.draw_networkx_edges(G, pos, width=1.0, alpha=0.7)
nx.draw_networkx_labels(G, pos, font_size=12, font_color='black')


plt.title('MNIST Label Similarity Network Graph (Cosine Similarity)')
plt.show()
```



In [ ]:
```python
# NetworkX graph oluştur
G = nx.Graph()

# Düğümleri ekle
for i in range(10):
    G.add_node(i, label=str(i))

# Kenarları ekle (benzerlik eşiği belirleyerek)
threshold = 4  # Benzerlik eşiği, bunu ayarlayabilirsiniz
for i in range(10):
    for j in range(i + 1, 10):
        if euclidean_distances[i, j] >= threshold:
            G.add_edge(i, j, weight=euclidean_distances[i, j])


# Graph'ı çiz
plt.figure(figsize=(10, 8))
pos = nx.spring_layout(G, seed=42)  # Düğüm yerleşimini belirle
nx.draw_networkx_nodes(G, pos, node_size=700, node_color='lightblue')
nx.draw_networkx_edges(G, pos, width=1.0, alpha=0.7)
nx.draw_networkx_labels(G, pos, font_size=12, font_color='black')


plt.title('MNIST Label Similarity Network Graph (Euclidean Distance)')
plt.show()
```

MNIST Label Similarity Network Graph (Euclidean Distance)