4 Task4

5 Task4.a

The code of new index compution:

```
def Gauss_sim_2d(P1,P2,gamma=1.0):
    dis=np.linalg.norm(P1-P2)
    return np.exp(-gamma * dis ** 2)

def new_validation_index(data,pred_labels):
    new_data=data.to_numpy().astype("float64")
    lens=len(new_data)
    T=0
    for i in range(lens):
        a,b=0,0
        for j in range(lens):
            if j!=i:
                sim=Gauss_sim_2d(new_data[i],new_data[j])
                a+=(pred_labels[i]==pred_labels[j])*sim
                b+=sim
        T+=a/b
    T=T/lens
    return T
```

Table 5: Comparison of 3 clustering methods ($K = [2, 5]$)

method	New Index			
K=	2	3	4	5
K-means	0.977	0.946	0.95	0.943
SC (Gaussian)	1.0	1.0	0.9965	0.9921
SC (Laplacian)	0.9782	0.975	0.9663	0.9592

Spectral clustering with RBF are still best.

5.1 Task4.b

Comparing with SI and DB index above, we can see τ index is a better choice in this spiral case. Using Spectral clustering with RBF when $K == 3$ obtain $\tau = 1.0$. But accordingly, imprecise results obtained using other methods also yielded better values with this validation method. Because this validation method currently considers only distance-based similarity. So as long as the points that are as close as possible are grouped in the same cluster, better index values can be calculated. The non-convex case of clusters is not taken into account.

5.2 Task4.c

Following the hint in the description, I took the k-nearest-neighbour graph adjacency matrix into consideration as well. Instead of calculating the similarity of all points and checking if they are in the same cluster, we now calculate the similarity if the points considered as neighbors are in the same cluster or not. The number of the nearest neighbors as a basis is a hyperparameter to be considered. The formula are below:

$$\tau = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum^{j\in neighbour} c_{ij}s(X_i,X_j)}{\sum^{j\in neighbour} s(X_i,X_j)} \tag{1}$$

Based on the index above, we obtain all $\tau = 1.0$ when it performed on Spectral Cluster with Gaussian kernel and $K = 3$, the number of nearest neighbours are 5,10,20. But under the same conditions $SI = 0.001$ and $DB = 5.882$, which means almost failed classification results based on these two indexes. But In fact The performance of spectral sluster with gaussian kernel are pretty good from the graph. If all points compute their neighbors in the same clusters as themselves as much as possible, naturally this classification result performs well under this validation metric. The disadvantage of this metric is due to the need to calculate the number of k nearest neighbors, which is a hyperparameter that makes it difficult to determine an optimal value.

The code are below:

```
def indexs_using_neighbour(data,pred_labels,n_nghbours=20):

    kg=kneighbors_graph(sprialsdata_train,n_neighbors=
       n_nghbours)

    new_data=data.to_numpy().astype("float64")
    lens=len(new_data)
    T=0
    for i in range(lens):
        a,b=0,0
        for j in range(lens):
            if j!=i:
                sim=Gauss_sim_2d(new_data[i],new_data[j])
                a+= kg[i,j] *(pred_labels[i]==pred_labels[j])*
                   sim
                b+=kg[i,j] * sim
        T+=a/b
    T=T/lens
    return T
```

References

[1] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[2] D. Davies and D. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.