Runheng Xu

xurunheng

[xu.runh@northeastern.edu](mailto:xu.runh@northeastern.edu)

Question 1

1 <https://colab.research.google.com/drive/1A8y7u-ohjgowArqXdhFdL8TxbZgNQqc5#scrollTo=BBQwc_NpXG_x>

def kmeans(dset, k=2, tol=1e-4):

    '''

    K-means implementationd for a

    `dset`:  DataFrame with observations

    `k`: number of clusters, default k=2

    `tol`: tolerance=1E-4

    '''

    # Let us work in a copy, so we don't mess the original

    working\_dset = dset.copy()

    # We define some variables to hold the error, the

    # stopping signal and a counter for the iterations

    err = []

    goahead = True

    j = 0

    # Step 2: Initiate centroids

    centroids = initiate\_centroids(k, dset)

    for \_ in range(100):

        # Step 3 and 4 - Assign centroids and calculate error

        working\_dset['centroid'], j\_err = centroid\_assignation(working\_dset, centroids)

        err.append(sum(j\_err))

        # Step 5 - Update centroid position

        centroids = working\_dset.groupby('centroid').agg('mean').reset\_index(drop = True)

    working\_dset['centroid'], j\_err = centroid\_assignation(working\_dset, centroids)

    centroids = working\_dset.groupby('centroid').agg('mean').reset\_index(drop = True)

    return working\_dset['centroid'], j\_err, centroids

2

fig, ax = plt.subplots(figsize=(8, 6))

plt.scatter(df.iloc[:,0], df.iloc[:,1],  marker = 'o',

            c=df['centroid'].astype('category'),

            cmap = customcmap, s=80, alpha=0.5)

plt.scatter(centroids.iloc[:,0], centroids.iloc[:,1],

            marker = 's', s=200, c=[0, 1, 2,3,4],

            cmap = customcmap)

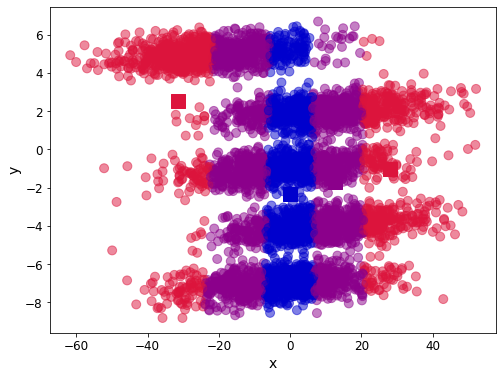
ax.set\_xlabel(r'x', fontsize=14)

ax.set\_ylabel(r'y', fontsize=14)

plt.xticks(fontsize=12)

plt.yticks(fontsize=12)

plt.show()



3 as we can see there are 5 cluster in dataset, the final result doesn’t’ cluster very well, if we initialize centroid to the center what we observed, the result would be better

Question2