Data Mining (Δ02): Exercise Set 2: 2.2 - 4Rectangles Dataset

Name: Nefeli Eleftheria Sextou

Student ID: 503

E-mail: pcs00503@uoi.gr, nsekstou@cs.uoi.gr

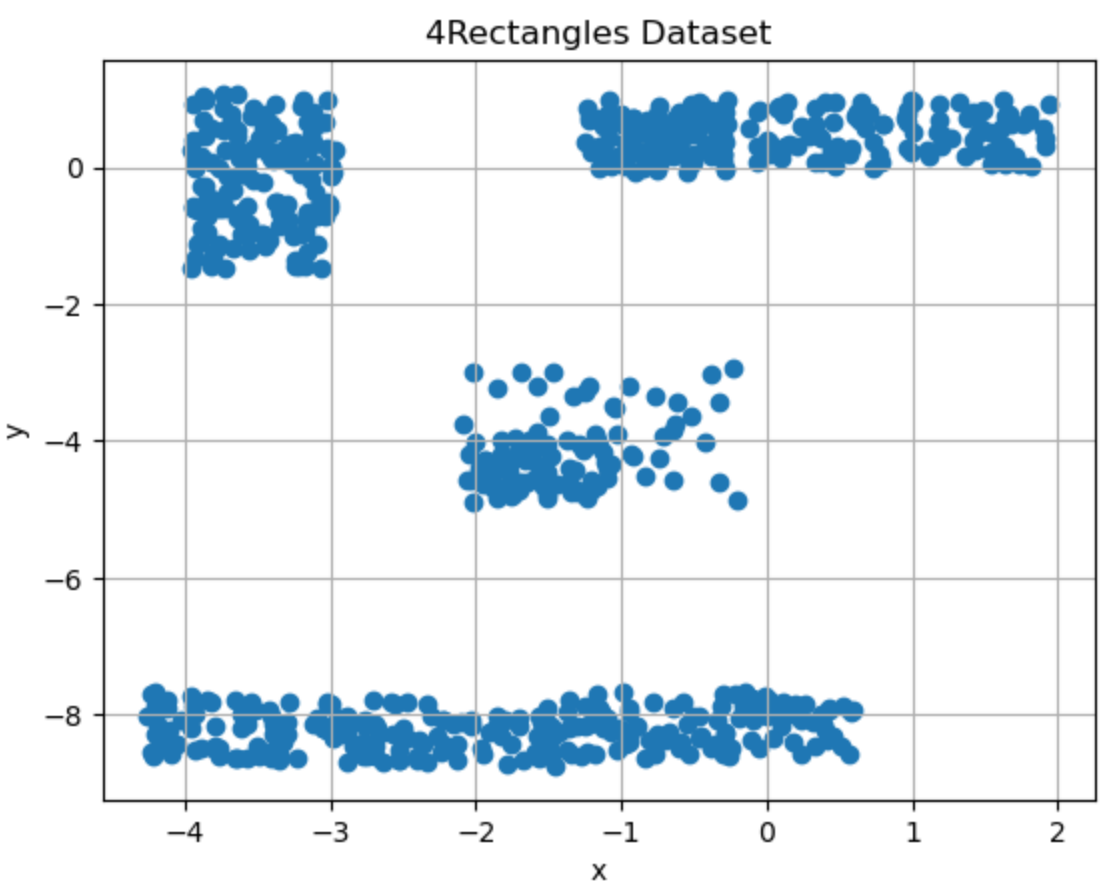
In [1]:
```
#general
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#data preprocessing


#classifiers
from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import SpectralClustering

#to ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

Load Data

In [2]:
```
data = []
# read the file line by line
with open(r'C:\Users\Nefeli\Desktop\dm_msc\DM_Homework2_2024\clustering\4rectangles.txt'
    for line in file:
        # Strip whitespace, split (by space)
        clean_line = line.strip().split()
        data.append((float(clean_line[0]), float(clean_line[1])))

# Create DataFrame from the list
main_df = pd.DataFrame(data, columns=['x', 'y'])
#rings3
```

Plot data

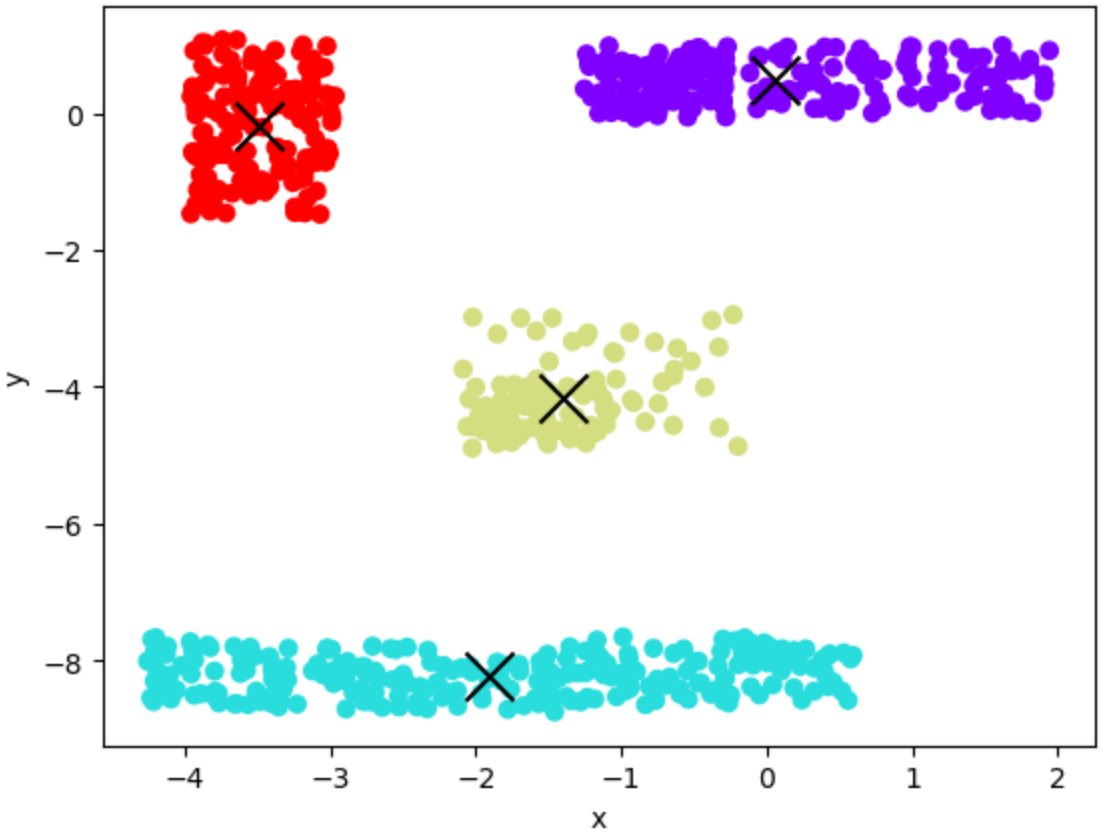
In [3]:
```
plt.scatter(main_df['x'], main_df['y'])
plt.xlabel('x')
plt.ylabel('y')
plt.title('4Rectangles Dataset')
plt.grid(True)
plt.show()
```



Expected number of clusters: 4

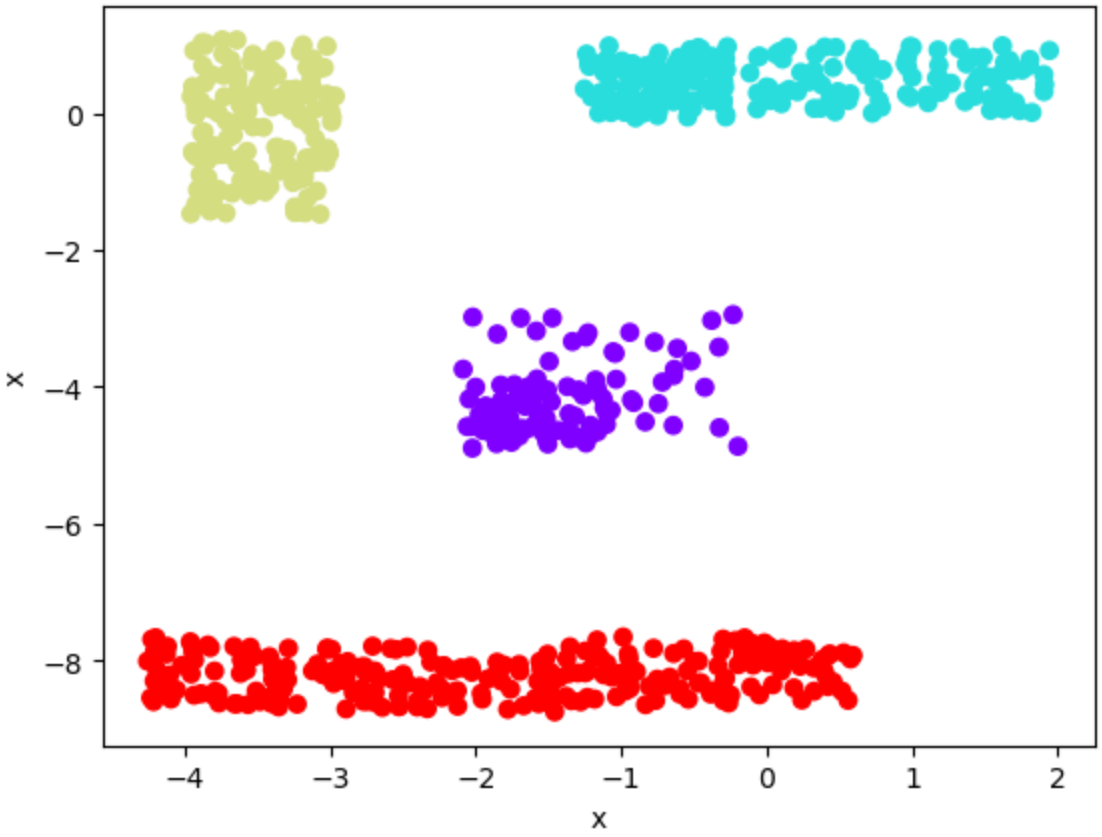
k-means

In [4]:
```
#init and fit
kmeans = KMeans(n_clusters=4)
kmeans.fit(main_df)

# get centroids and labels
centroids = kmeans.cluster_centers_
labels = kmeans.labels_

plt.scatter(main_df['x'], main_df['y'], c=labels, cmap='rainbow')
plt.scatter(centroids[:, 0], centroids[:, 1], s=300, c='black', marker='x')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



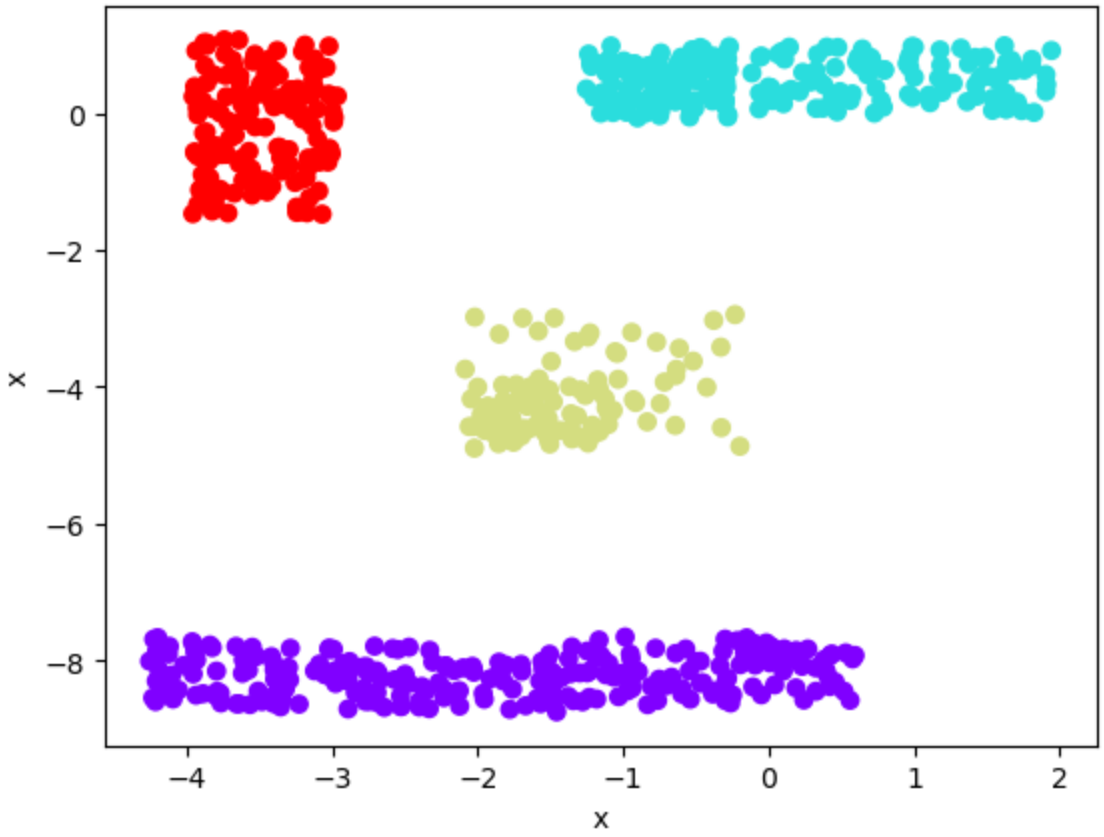
Agglomerative Clustering : single link

In [5]:
```
#init and fit
agg_cluster = AgglomerativeClustering(n_clusters=4, linkage='single')

#get labels
labels = agg_cluster.fit_predict(main_df)

plt.scatter(main_df['x'], main_df['y'], c=labels, cmap='rainbow')
plt.xlabel('x')
plt.ylabel('x')
plt.show()
```



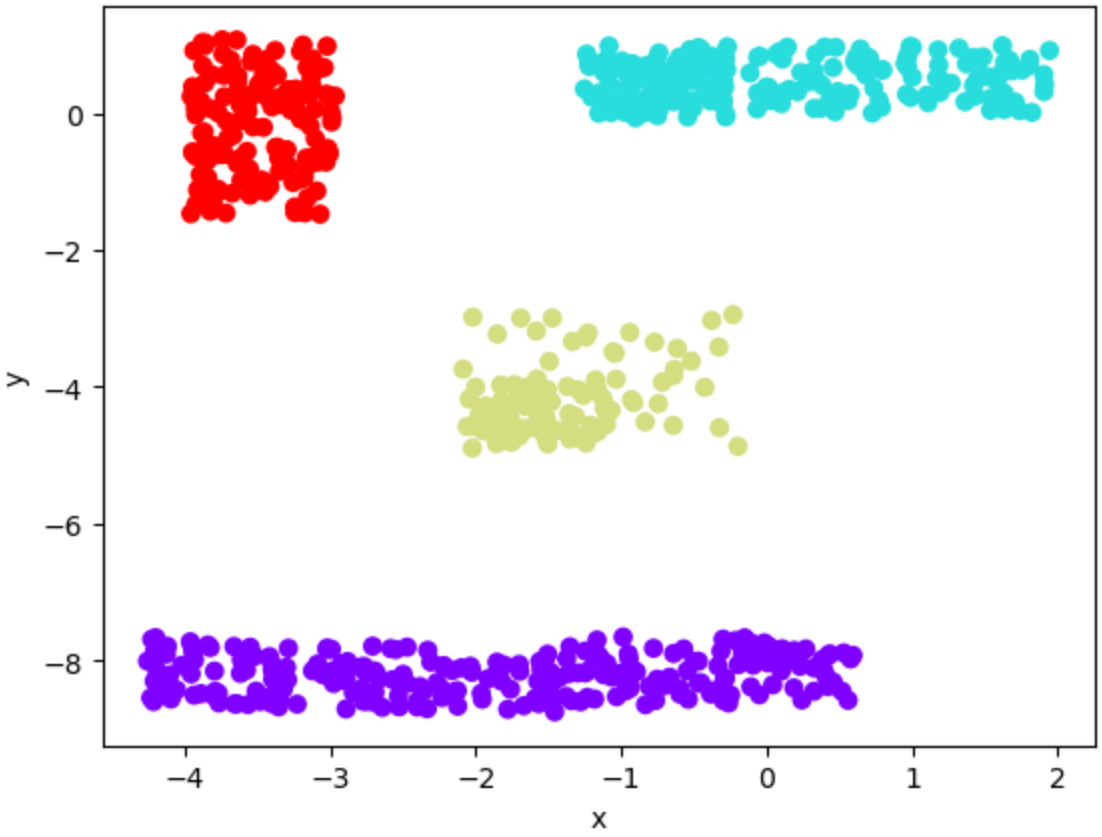
Agglomerative Clustering : average link

In [6]:
```
#init and fit
agg_cluster = AgglomerativeClustering(n_clusters=4, linkage='average')

#get labels
labels = agg_cluster.fit_predict(main_df)

plt.scatter(main_df['x'], main_df['y'], c=labels, cmap='rainbow')
plt.xlabel('x')
plt.ylabel('x')
plt.show()
```



Spectral Clustering

In [7]:
```
# perform spectral clustering
sigma = 0.5 # tried 0.1, 0.5, 1
spectral_cluster = SpectralClustering(n_clusters=4, affinity='rbf', gamma = (1/(sigma**2
labels = spectral_cluster.fit_predict(main_df)

plt.scatter(main_df['x'], main_df['y'], c=labels, cmap='rainbow')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



Remarks

All classifiers succeed in producing the expected clustering result. This is because the clusters are well separated, have similar shapes and densities and overall low variability. This enables all classifiers applied, regardless of their properties to achieve a good result.

In [ ]: