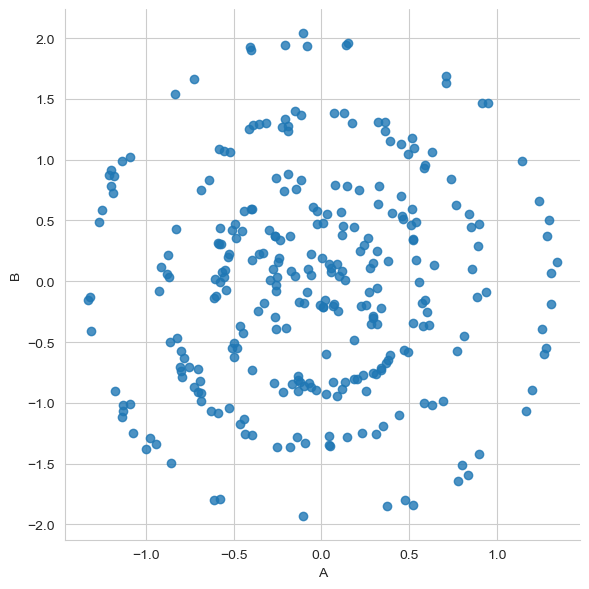
Name: Karteek Swasaka  
Mobile: 9494846632  
  
  
1. Importing mandatory libraries  
  

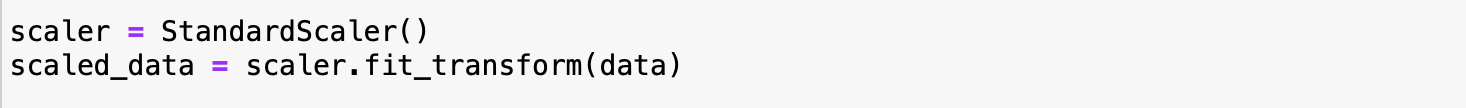

2. Reading the data from the given csv file  
  


3. Data points in scatter plot looks like this  
  


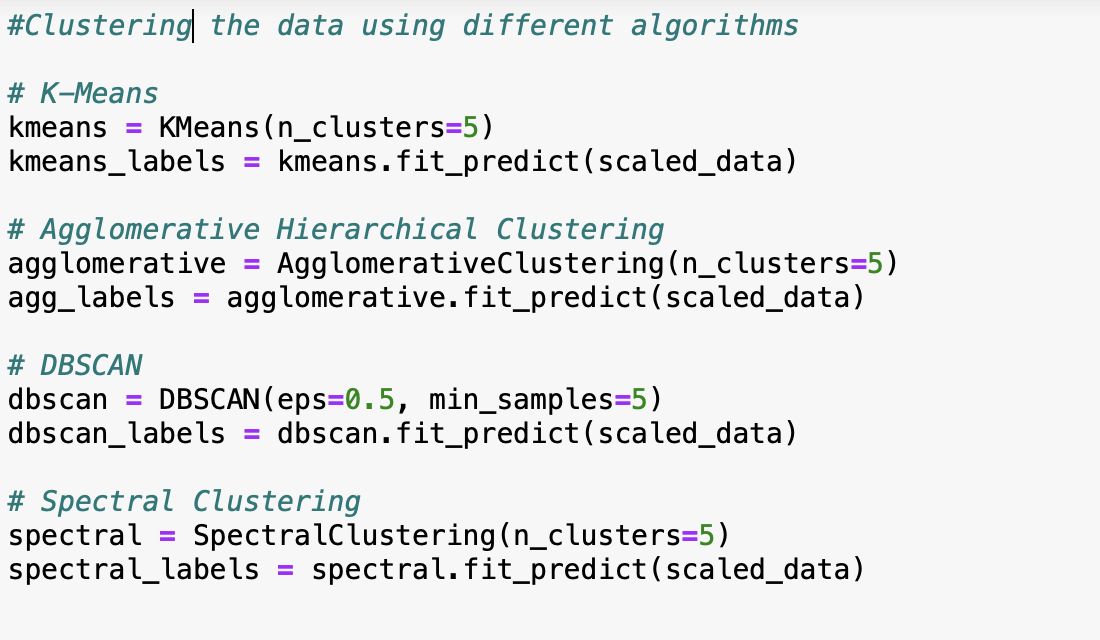
4. Checking the null values



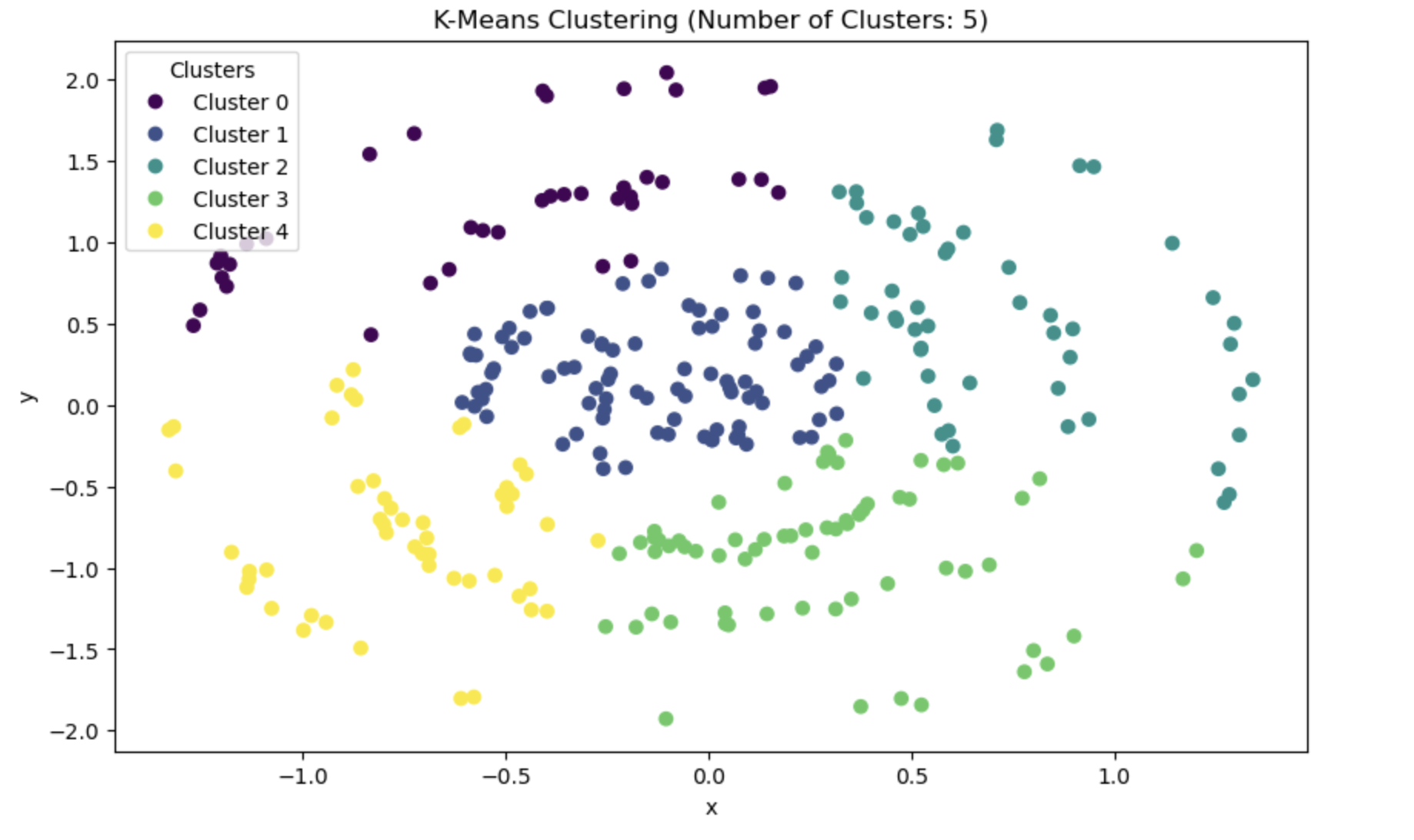
Which means, no Null values in the DataFrame

5. Scaling the values using Standard Scalar  
  


6. Performing Different clustering algorithms on the given data



7. **Clusters with K- Means Algorithm**



K-Means Silhouette Score: 0.36429898828652335

Cluster Sizes:

Cluster 0: 39 points

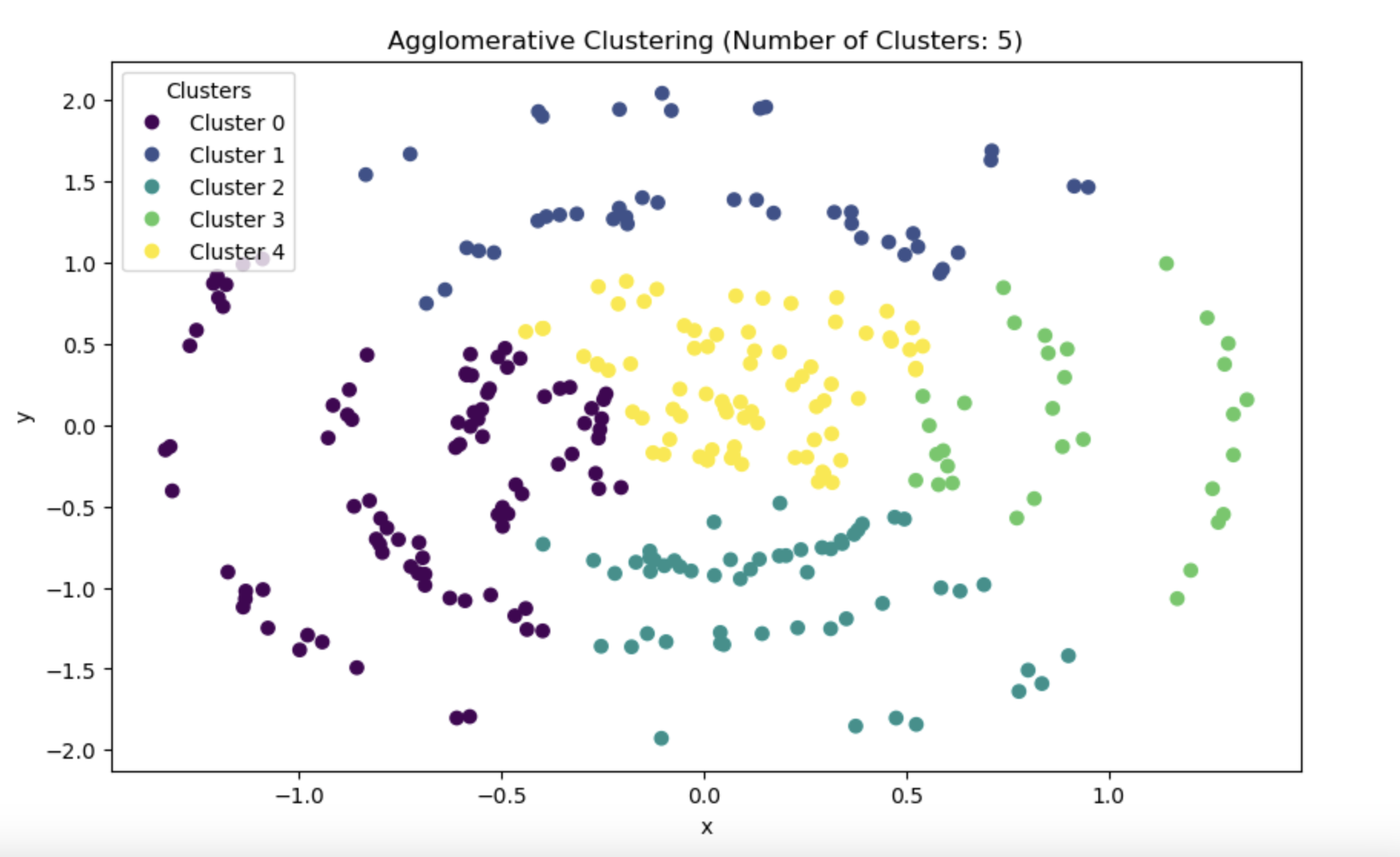
Cluster 1: 88 points

Cluster 2: 52 points

Cluster 3: 65 points

Cluster 4: 51 points

**8. Agglomerative Hierarchical Clustering**

  
Agglomerative Clustering Silhouette Score: 0.3122779541990143

Cluster Sizes:

Cluster 0: 90 points

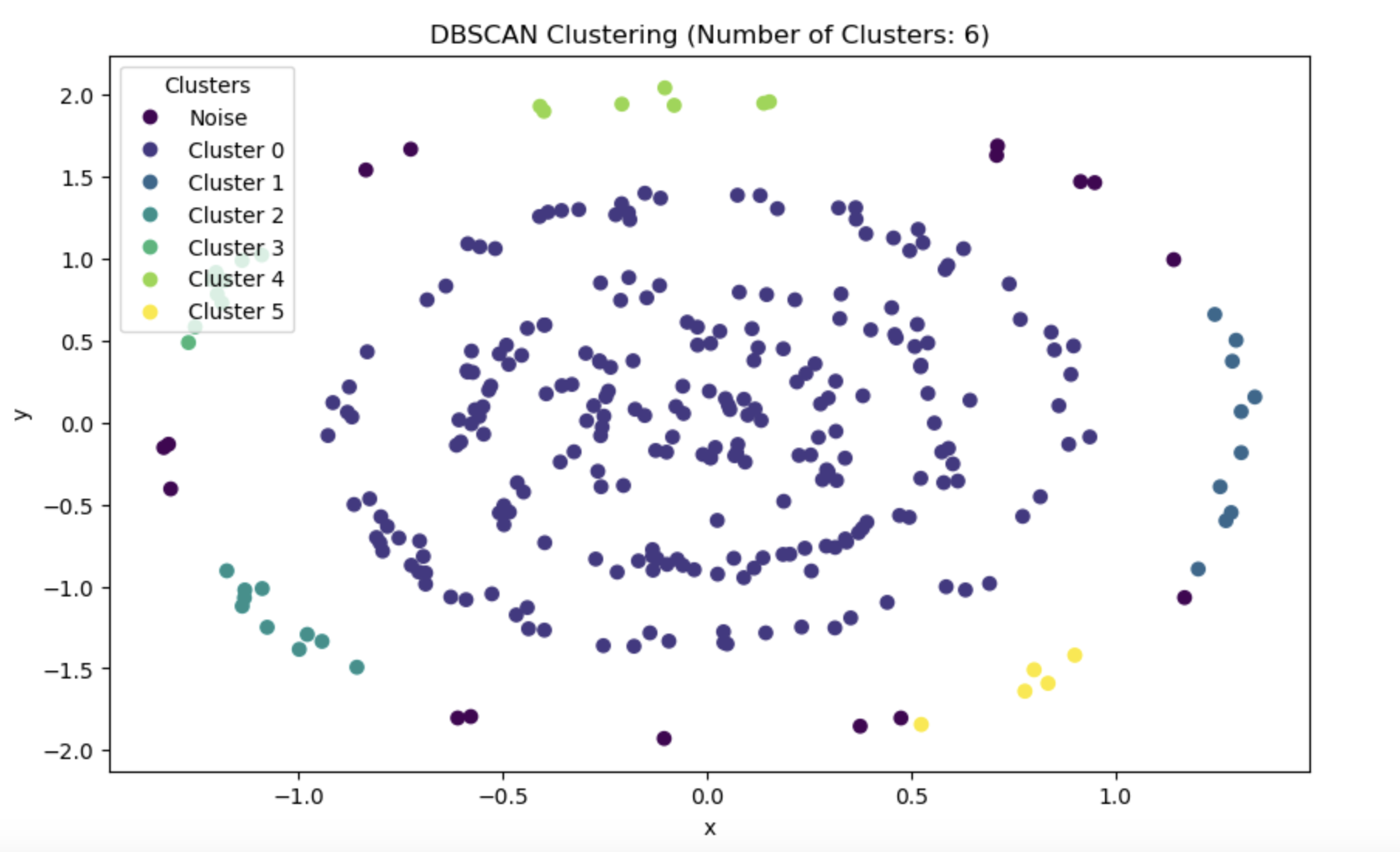
Cluster 1: 42 points

Cluster 2: 55 points

Cluster 3: 32 points

Cluster 4: 76 points

9. **DBSCAN**



Cluster Sizes:

Noise: 16 points

Cluster 0: 238 points

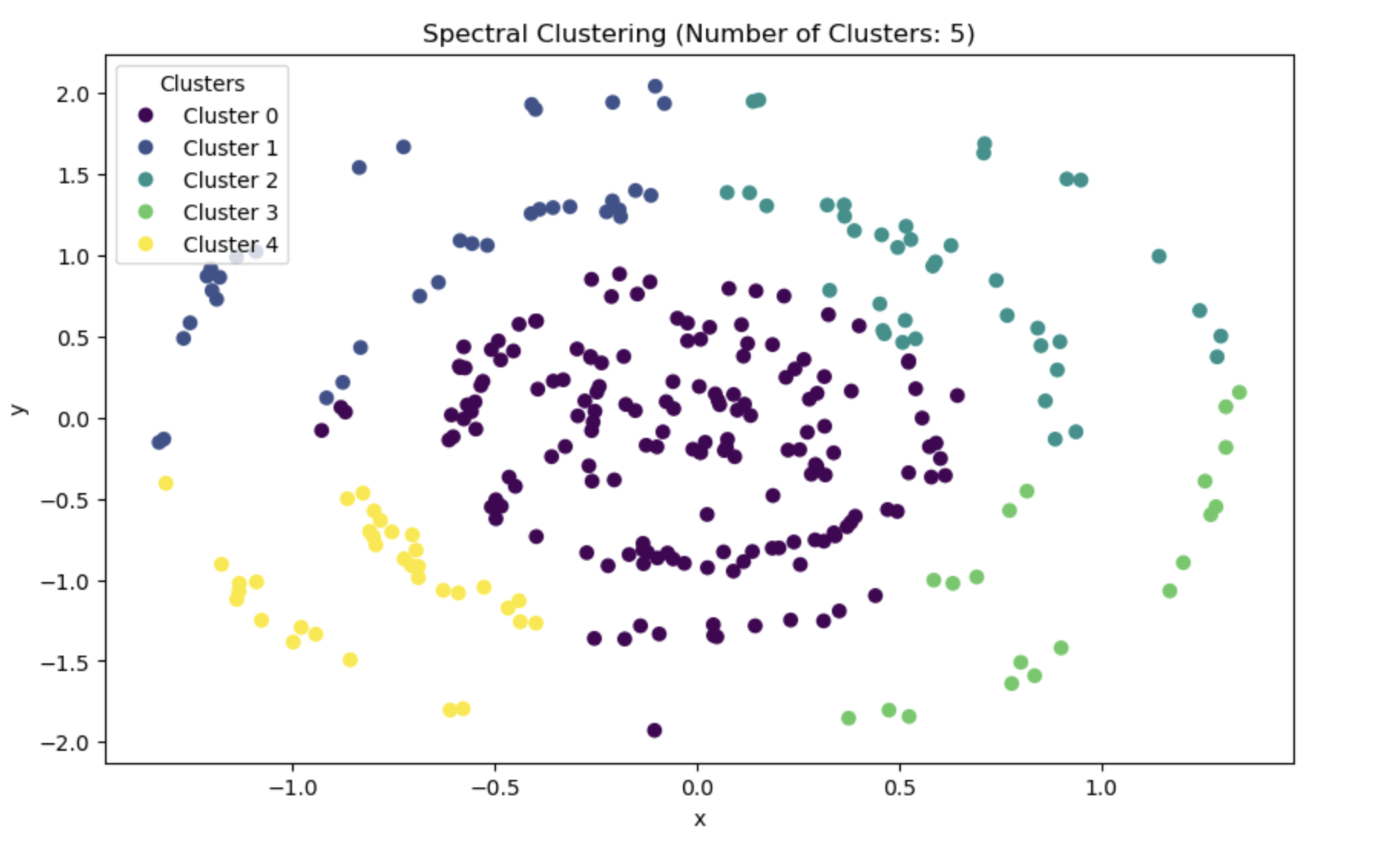
Cluster 1: 10 points

Cluster 2: 10 points

Cluster 3: 9 points

Cluster 4: 7 points

Cluster 5: 5 points

10. **Spectral Clustering**  
  


Cluster Sizes:

Cluster 0: 165 points

Cluster 1: 36 points

Cluster 2: 40 points

Cluster 3: 20 points

Cluster 4: 34 points

**My Choice of clustering algorithm**

If we compare the silhouette scores of all 4 algorithms. K Means is the best. But   
If your data points form circular or spherical clusters in a 2D or 3D space, one of the best clustering algorithms to consider is DBSCAN (Density-Based Spatial Clustering of Applications with Noise). DBSCAN is effective in detecting clusters of arbitrary shapes, including circular and spherical clusters.   
  
**So here My Choice is DBSCAN**  
  
Here's why DBSCAN is a good choice for such data  
  
**Cluster Shape Flexibility:** DBSCAN can identify clusters of varying shapes, including circular and spherical clusters. It does not make assumptions about the shape of clusters, making it suitable for non-convex structures.

**Automatic Density Estimation:** DBSCAN automatically determines the number of clusters based on the data's local density. This is beneficial when you don't know the exact number of circular clusters in advance.

**Noise Handling:** DBSCAN is robust in handling noise points (outliers) and does not force every point to belong to a cluster. Noise points are labeled as outliers, which is useful when dealing with data imperfections.

**No Need for Predefined Radius:** Unlike some other methods that require you to specify the radius or size of clusters, DBSCAN uses a parameter (eps) to define a distance threshold, which adapts to the local density of the data.

**Scalability:** DBSCAN is computationally efficient and can handle large datasets, making it suitable for clustering circular clusters in a high-dimensional space.

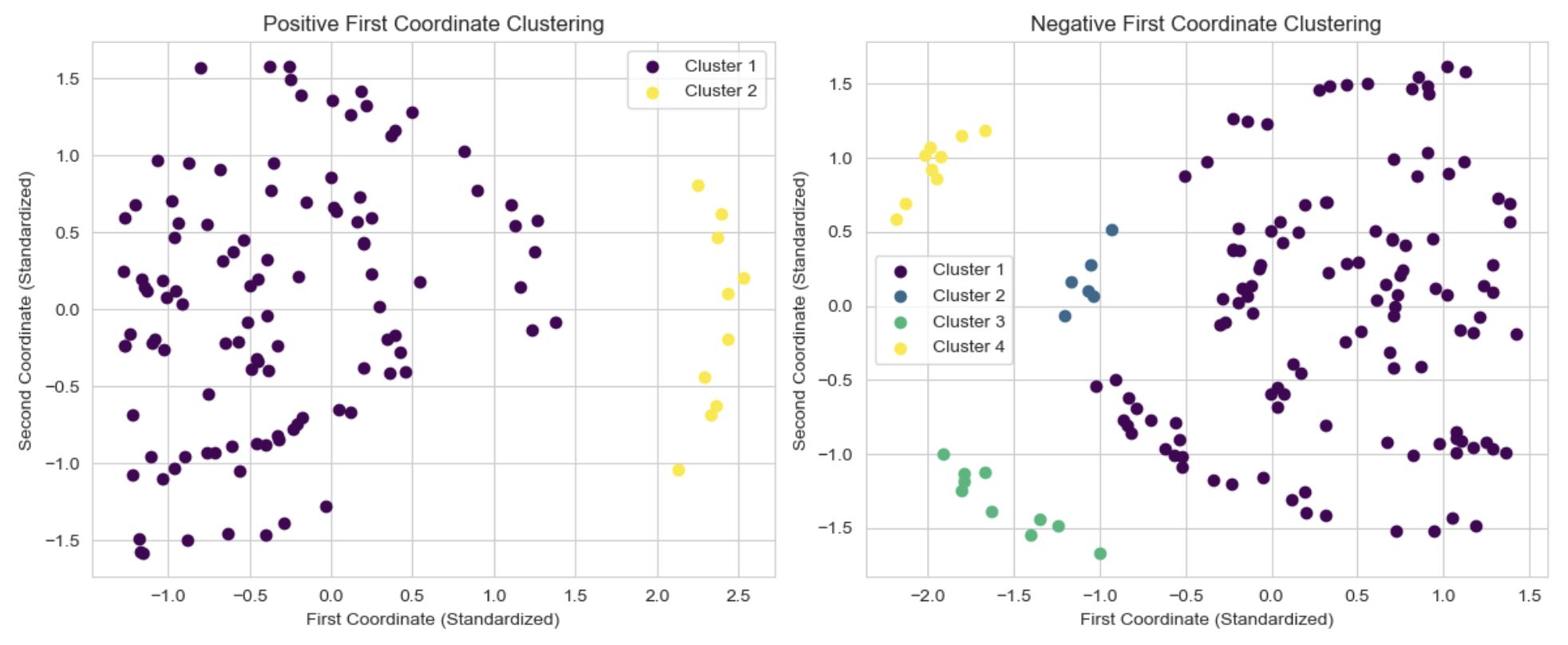
**2nd Part of the assessment**

"Negative First Coordinate": This refers to points where the x-coordinate is negative. These points fall in Quadrant II (where x is negative and y is positive) or Quadrant III (where both x and y are negative) in a 2D graph. So, need should consider points from both Quadrant II and Quadrant III.

"Positive First Coordinate": This refers to points where the x-coordinate is positive. These points fall in Quadrant I (where both x and y are positive) or Quadrant IV (where x is positive and y is negative) in a 2D graph. So, need to consider points from both Quadrant I and Quadrant IV.

In summary, when you are asked to consider "negative first coordinate" and "positive first coordinate," you should take points from both the left side of the Cartesian plane (Quadrants II and III) for "negative first coordinate" and points from both the right side of the Cartesian plane (Quadrants I and IV) for "positive first coordinate."

With the DBSCAN clustering algorithm, clusters containing points with negative first coordinates should not be grouped together with clusters containing points with positive first coordinates.



Positive First Coordinate Clustering:

Number of Clusters: 2

Cluster Sizes:

Cluster 1: 108 points

Cluster 2: 10 points

Negative First Coordinate Clustering:

Number of Clusters: 4

Cluster Sizes:

Cluster 1: 116 points

Cluster 2: 6 points

Cluster 3: 10 points

Cluster 4: 9 points