Analysis of PROBLEM 2 results

General DBSCAN Observations:

- DBSCAN's performance is highly dependent on the eps and min_samples parameters.
- An optimal balance between these parameters is crucial for identifying meaningful clusters without excessive noise.
- We may want to consider visual inspection of clustering results to assess the effectiveness of clustering better

From question (a), I found that DBSCAN automatically identifies outliers during clustering, excels in identifying clusters of varying shapes and sizes.

But The performance of DBSCAN heavily relies on the choice of parameters eps(the maximum distance between two samples for one to be considered as about the other) and min_samples (the minimum number of points required to form a dense region). Poor parameter choices can lead to either too many noise points or insufficiently defined clusters.

Also, it can struggle with high-dimensional data, making it less efficient.

K-Means is simple to implement and computationally efficient, making it a go-to algorithm for many clustering tasks, especially for large datasets. Each cluster is represented by a centroid, providing a clear, interpretable summary of each cluster's characteristics.

But K-Means assumes that clusters are spherical (i.e., equally sized and shaped). This makes it less effective for datasets with irregularly shaped clusters. The final clustering result can depend on the initial selection of centroids.

General Spectral Clustering Observations:

- The choice of the gamma parameter significantly impacts the clustering results, with lower values potentially merging distinct clusters and higher values possibly leading to overfitting.
- The RBF kernel allows the clustering algorithm to capture non-linear relationships, which can be beneficial in datasets where clusters are not linearly separable.

For example, with gamma = 1.0, the RBF kernel will consider neighbors within a certain distance, balancing the influence of nearby points while still allowing for some separation between clusters.

If the clusters are well-separated, it indicates that the data's inherent structure is being captured effectively.
With gamma = 0.5, Lowering gamma to `0.5` broadens the influence of each point, allowing more distant points to influence the cluster formation. This can lead to larger, more ambiguous clusters.
The risk here is that distinct clusters might merge into a single larger cluster due to the increased reach of the RBF kernel.

With gamma = 2.0, A higher gamma value (e.g., `2.0`) makes the clustering more sensitive to local structures. Points need to be very close to each other to be grouped, potentially resulting in small, well-defined clusters.
This sensitivity can lead to overfitting, where the algorithm creates clusters around noise or outliers rather than capturing the overall data distribution.
I may also want to try other affinity options or distance metrics in the Spectral Clustering algorithm, which could yield different clustering outcomes.

Visualizations are crucial to assess how well the clusters formed align with the actual data structure.