DBCAN

Daniel Ching, Mateen Shagagi

Objective: The purpose of this project was to implement the DBSCAN clustering algorithm from scratch using Java. The aim was to gain a deeper understanding of the algorithm's mechanics by coding it manually and applying it to a dataset to observe its behavior and effectiveness in real-world clustering tasks.

Core Concepts of DBSCAN:

- Core Points: a point that has a specified number of points (MinPts) within a given radius (ε)
- Border Points: points that are not core points but are close enough to core points to be considered part of a cluster
- Noise Points: Points that are neither core points nor border points. These are typically considered outliers in the data

How DBSCAN Works:
DBSCAN begins with an arbitrary starting point that has not been visited. The neighborhood of this point is extracted using a user-defined distance ε. If this neighborhood contains sufficiently many points (MinPts), a cluster is started. Otherwise, the point is labeled as noise. This process continues until all points have been processed.

Evaluating the Model and Tuning Hyperparameters:
We conducted two distinct attempts to evaluate and fine-tune the hyperparameters. These attempts were aimed at optimizing the clustering results through different evaluation metrics.

1. Reducing Max Distance Between Points

   We focused on minimizing the maximum distance between points within each cluster. This metric was chosen with the intent to enhance cluster compactness, ensuring that all points within a cluster are closely knit, thus potentially increasing the homogeneity of the clusters. This method often led to over-segmentation, where the data was divided into too many small clusters, complicating the interpretability of the results.

2. Silhouette Score

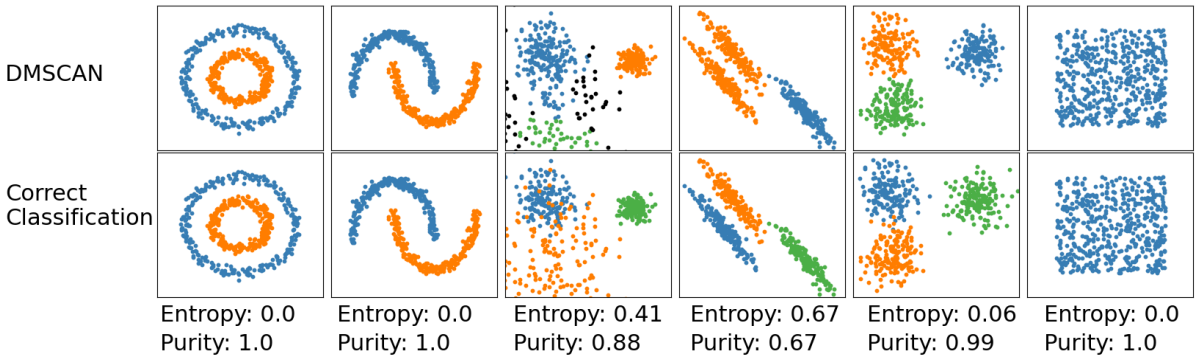
   We shifted to using the Silhouette Score for our second evaluation attempt. The Silhouette Score is a well-regarded metric that measures both the cohesion within clusters and the separation between different clusters, offering a more balanced view of cluster quality. This method not only considered the compactness of clusters but also their distinctness from each other.

Dataset:
For our project, we utilized the Mall Customer Segmentation Dataset, a popular dataset often used in marketing and customer segmentation studies. This dataset provides a good basis for testing clustering algorithms due to its variety of features and the meaningful insights it can yield about consumer behavior.

Features:

- ID: A unique identifier for each customer
- Age: The age of the customer
- Gender: The gender of the customer, which is categorized as male or female.
- Income: The annual income of the customer, expressed in thousands of dollars.
- Spending Score: A score assigned by the mall based on customer behavior and spending nature. The score ranges from 1 to 100, with higher scores indicating higher spending.

Results:
Overall, our model worked well with non-circular cluster assignments as was expected with a DBSCAN implementation. Our evaluator using the silhouette score proved to be quite accurate when comparing our results with the human classified data.



For the 2-dimensional data, 4/6 of the datasets we used to test seemed to almost perfectly be assigned clusters. The other 2 proved to be difficult given the way the points are positioned.

For the 3-dimensional data (customer dataset), our model did decent at finding the main customer groups, however assigned many points as noise due to the differing densities within the clusters.