Clustering Algorithms – K-Means and DBSCAN

Understanding Unsupervised Learning Techniques

Ankit Shrestha

What is Clustering?

Clustering is an unsupervised learning technique that groups similar data points into clusters.

Goal: Maximize similarity within clusters and minimize similarity between clusters.

Use Cases:

- Market segmentation
- Image compression
- Customer behavior analysis
- Anomaly detection

Types of Clustering

- Partition-based: e.g., K-Means
- Density-based: e.g., DBSCAN
- Hierarchical: Builds nested clusters (tree-like structure)

K-Means Clustering

Concept: Divides data into *K* clusters based on distance to cluster centers (centroids).

Algorithm Steps:

1. Choose number of clusters (*K*).
2. Initialize *K* centroids randomly.
3. Assign each data point to the nearest centroid.
4. Recalculate centroids (mean of assigned points).
5. Repeat until centroids stabilize.

Distance Metric: Usually Euclidean distance.

Choosing Optimal K

- Elbow Method: Plot K vs. inertia (sum of squared distances); choose point where curve bends.

- Silhouette Score: Measures how well points fit in their clusters (range: -1 to 1).

Advantages & Limitations

Advantages:

- Simple and fast
- Works well with large datasets

Limitations:

- Requires predefining *K*
- Sensitive to outliers
- Assumes spherical clusters

DBSCAN Clustering

Concept: Groups together points that are closely packed and marks points in low-density regions as outliers.

Key Parameters:

- `eps`: maximum distance between two points to be considered neighbors
- `min_samples`: minimum number of points required to form a dense region

- Algorithm Steps:

  1. Pick an unvisited point.
  2. Find all nearby points within eps.
  3. If the number ≥ min_samples, form a cluster.
  4. Expand cluster by recursively including nearby points.
  5. Mark remaining points as noise if not part of any cluster.

Choosing DBSCAN Parameters

- Use k-distance graph: plot sorted distances of each point to its k-th nearest neighbor; look for sharp bend → eps.

- min_samples often ≈ dimensionality + 1.

Advantages & Limitations

Advantages:

- No need to specify number of clusters
- Detects arbitrary-shaped clusters
- Handles noise/outliers well

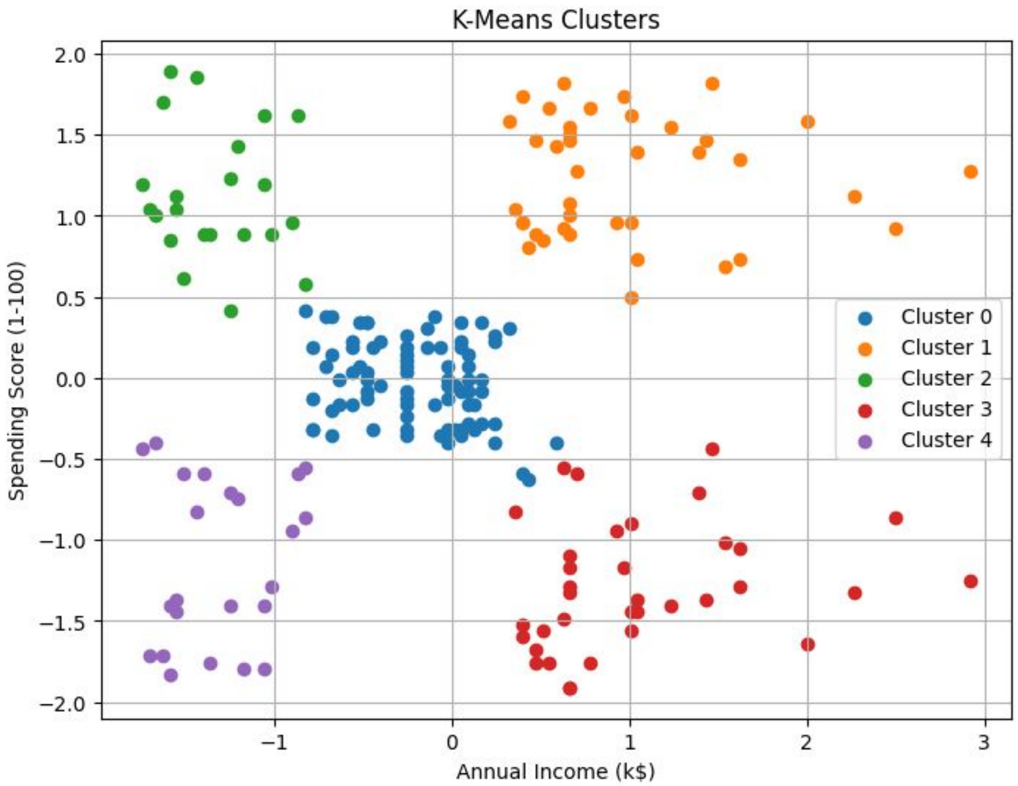
Limitations:

- Struggles with varying density clusters
- Parameter sensitivity (eps, min_samples)

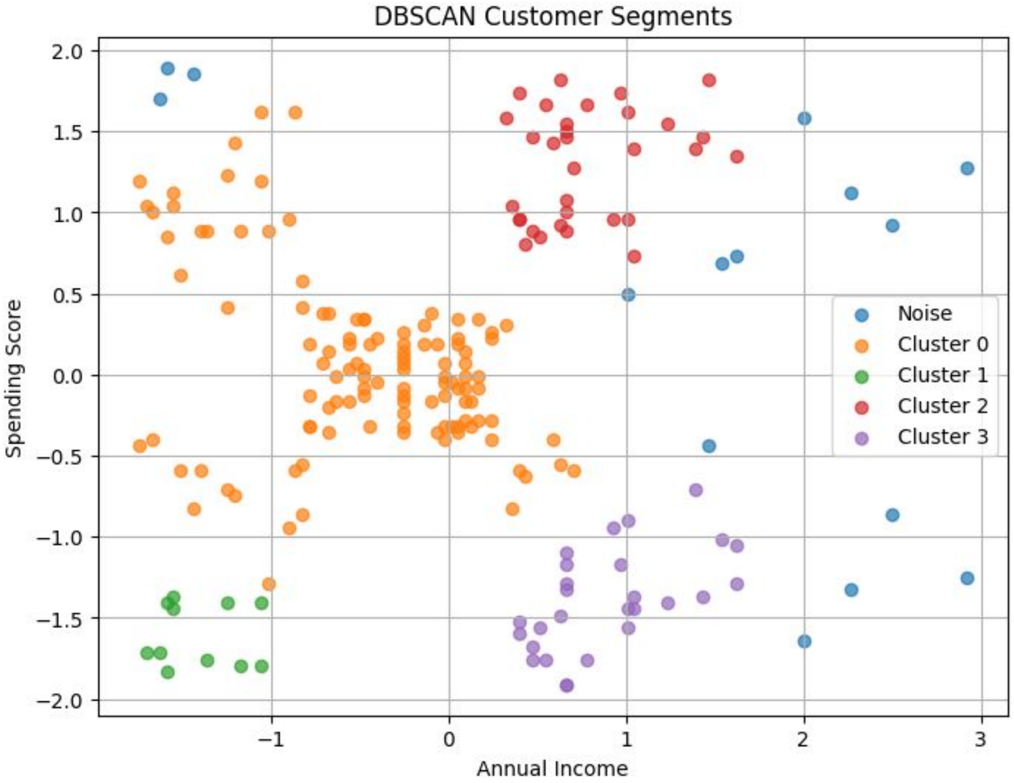
Comparison Summary

Feature	**K-Means**	**DBSCAN**
Cluster Shape	Spherical	Arbitrary
Outlier Handling	Poor	Good
Need for K	Yes	No
Scalability	High	Moderate
Density Sensitivity	Low	High

Scatter plot of K-Means



Scatter plot of DBSCAN



Applications

- K-Means: Customer segmentation, document clustering, color quantization
- DBSCAN: Fraud detection, spatial data (GPS), anomaly detection

Thank You !