MEAN SHIFT CLUSTERING ALGORITHM

Mean Shift:

- Mean Shift is a non-parametric clustering algorithm that doesn't require prior knowledge of the number of clusters in the data. which aim to locate the modes or peaks of a probability density function. The algorithm iteratively shifts data points towards the mode of the underlying data distribution until convergence.

- **How Does Mean Shift Work:**

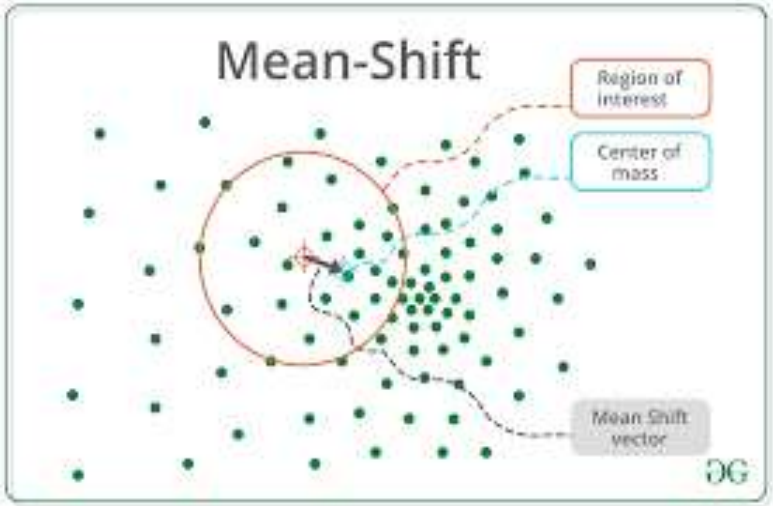
1. **Kernel Density Estimation**:
   1. Mean Shift starts by estimating the underlying probability density function of the data using kernel density estimation. This involves placing a kernel on each data point and summing up the contributions to estimate the density at each point.
2. **Mean Shift Operation**:
   1. In the mean shift operation, each data point is iteratively shifted towards the mode (or peak) of the estimated density function. This shift is determined by calculating the mean of the points within a certain neighborhood defined by a bandwidth parameter.
3. **Convergence**:
   1. The process continues until convergence, where data points settle into the modes of the density function. At convergence, each data point is assigned to the mode it converges to, thereby forming clusters.

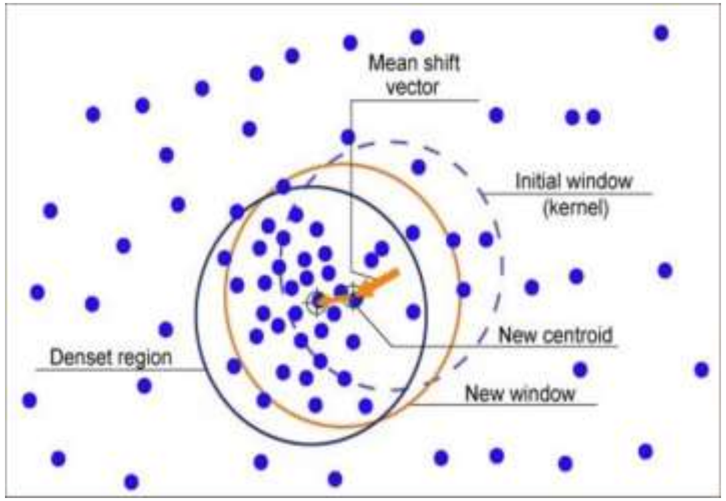
MEAN SHIFT CLUSTERING ALGORITHM

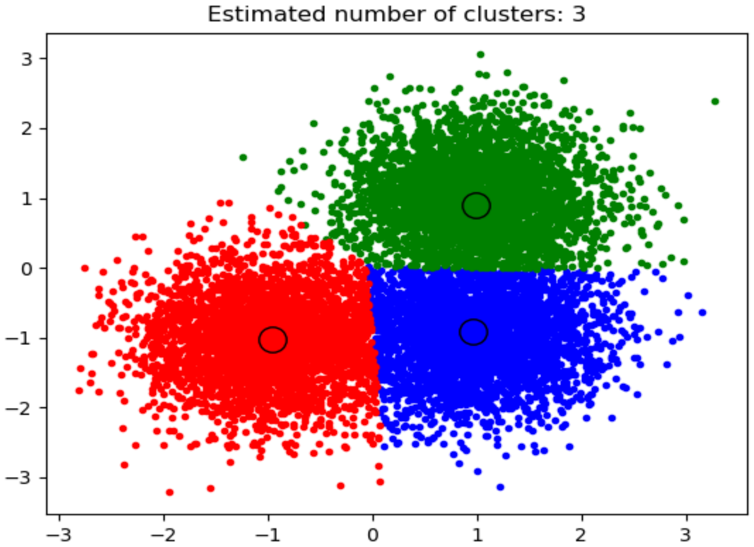
- **Why Use Mean Shift:**

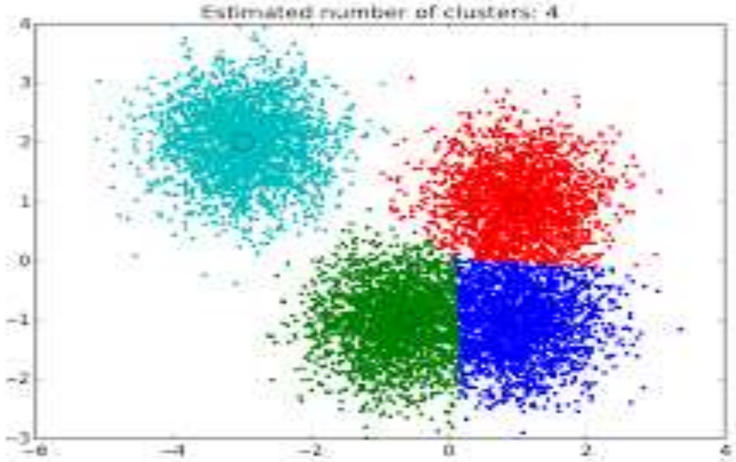
1. **No Predefined Number of Clusters:**
    1. Mean Shift doesn't require specifying the number of clusters beforehand, making it suitable for datasets where the number of clusters is not known a priori.
2. **Robust to Irregularly Shaped Clusters:**
    1. Mean Shift can identify clusters of arbitrary shapes and sizes. It adapts well to the underlying data distribution, making it robust in real-world scenarios.
3. **Automatic Mode Detection:**
    1. Mean Shift automatically detects the modes or peaks of the underlying data distribution, making it suitable for datasets with complex structures and multiple modes.

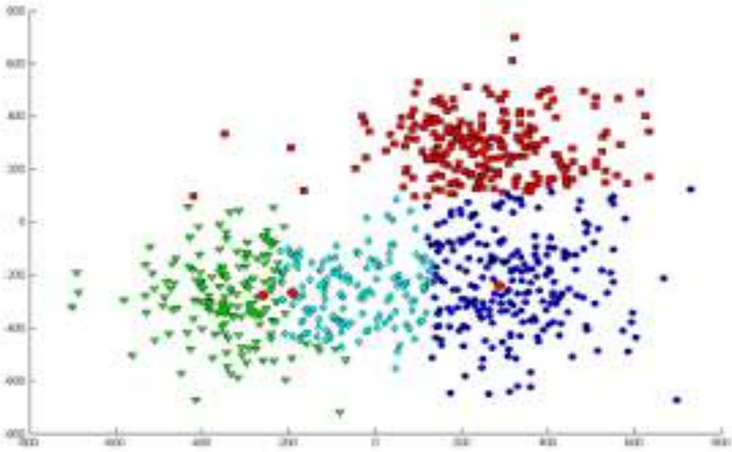
MEAN SHIFT CLUSTERING ALGORITHM

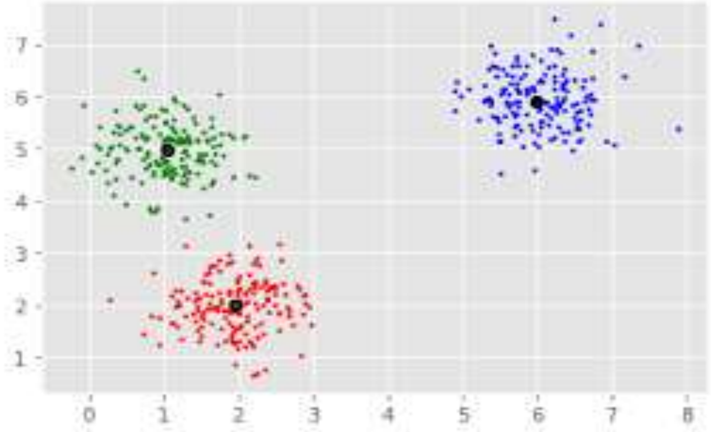












MEAN SHIFT CLUSTERING ALGORITHM

Let's break down the code step by step

```
from sklearn.cluster import MeanShift, estimate_bandwidth
from sklearn.datasets import make_blobs
```

These lines import necessary modules from scikit-learn: MeanShift for performing mean shift clustering, and estimate bandwidth for estimating the bandwidth of the kernel used in mean shift clustering. It also imports make_blobs to generate synthetic data for clustering

```
centers = [[1, 1], [-1, -1], [1, -1]]1]
```

This line defines the centers of the clusters. It creates a list where each element represents the coordinates of a cluster center.

```
X, _ = make_blobs(n_samples=2000, centers=centers, cluster_std=0.6) [-1, -1], [1, -1]]zzz
```

This line generates synthetic data points using the make_blobs function. It creates 2000 samples distributed around the specified cluster centers with a standard deviation of 0.6.

MEAN SHIFT CLUSTERING ALGORITHM

- **bandwidth = estimate_bandwidth(X, quantile=0.2, n_samples=2000)**
- This line estimates the bandwidth of the kernel used in mean shift clustering. Bandwidth is an important parameter that determines the size of the region to consider when performing mean shift clustering.
- ms = MeanShift(bandwidth=bandwidth, bin_seeding=True)
- This line initializes the MeanShift clustering algorithm with the previously estimated bandwidth. bin_seeding=True indicates that initial cluster centers are calculated using a binning technique.
- **ms.fit(X):**
- This line fits the mean shift clustering algorithm to the data X, identifying clusters and assigning labels to each data point.

MEAN SHIFT CLUSTERING ALGORITHM

- **labels = ms.labels_**
- This line retrieves the labels assigned to each data point after clustering.

- **cluster_centers = ms.cluster_centers_**
- This line retrieves the cluster centers computed by the mean shift algorithm.

- **labels_unique = np.unique(labels)**
- **n_clusters_ = len(labels_unique)**
- These lines calculate the number of unique clusters found by the mean shift algorithm.

- **print("number of estimated clusters : %d" % n_clusters_)**
- This line prints the number of estimated clusters found by the mean shift algorithm.

MEAN SHIFT CLUSTERING ALGORITHM

- **plt.figure(1)**
- **plt.clf()**
- **plt.title("Estimated number of clusters: %d" % n_clusters_)**
- This line sets up the figure for plotting and sets the title with the estimated number of clusters
- **colors = ["#dede00", "#377eb8", "#f781bf"] markers = ["+", "-", "*"]**
- These lines define colors and markers for different clusters to be used in the plot.
- **for k, col in zip(range(n_clusters_), colors):**
- **my_members = labels == k**
- **cluster_center = cluster_centers[k]**
- **plt.plot(X[my_members, 0], X[my_members, 1], markers[k], color=col)**
- **plt.plot(cluster_center[1], cluster_center[1], markers[k], markerfacecolor=col, markeredgecolor="k", markersize=50,)**
- This loop iterates over each cluster, plots the data points belonging to that cluster, and highlights the cluster centers.

MEAN SHIFT CLUSTERING ALGORITHM

- **plt.title("Estimated number of clusters: %d" % n_clusters_) plt.show()**
- the plot showing the estimated clusters and their centers.

- **Conclusion:**
- Spectral clustering is a versatile technique that addresses many of the limitations of traditional clustering algorithms. Its ability to handle complex datasets and capture non-linear relationships makes it a valuable tool in various domains.