# ML Clustering Algorithm

Unsupervised Learning

# INTRODUCTION

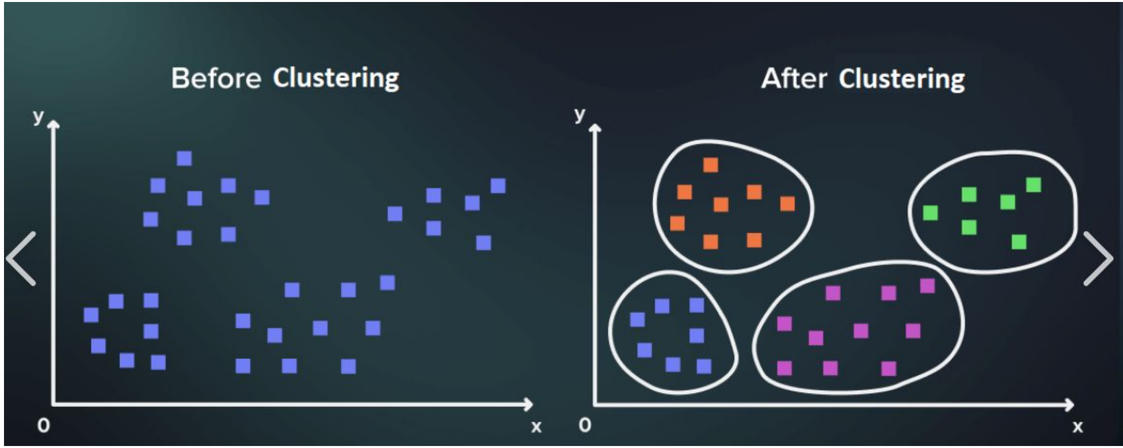
**What is Clustering?**

- Learning from an **unlabeled dataset** to identify patterns, groups, or clusters.
- **Output**: A set of clusters that group similar data points together.

**Key Applications**:

- Customer segmentation
- Market basket analysis
- Image segmentation

**Clustering is part of Unsupervised Learning**.



# Popular Clustering Algorithms

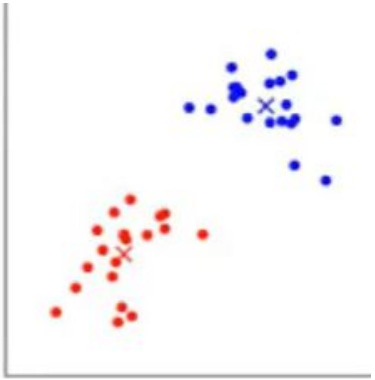
| Algorithm | Type | Key Features |
|---|---|---|
| **K-Means** | Partitional | Predefined clusters (K), groups data around centroids. |
| **Hierarchical Clustering** | Hierarchical | Builds clusters in a tree-like structure. |
| **Affinity Propagation** | Message Passing | Identifies exemplars, no predefined clusters. |
| **Mean Shift** | Density-Based | Groups points by moving toward density peaks. |
| **Spectral Clustering** | Graph-Based | Maps points to low-dimensional space for clustering. |
| **DBSCAN** | Density-Based | Groups dense areas; identifies outliers. |
| **OPTICS** | Density-Based | Handles clusters with varying densities. |
| **BIRCH** | Hierarchical | Summarizes large datasets for efficient clustering. |

# Assignment

**Problem Statement**:

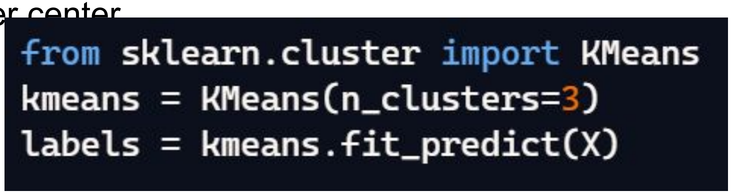
- A new customer joins. Predict which group or category they belong to so marketing campaigns or SMS messages can be customer-focused.

# K-Mean



**How It Works**:

1. Choose the number of clusters (K).
2. Assign each point to the nearest cluster center
3. Adjust centers and repeat until stable.



```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3)
labels = kmeans.fit_predict(X)
```

**Key Features**:

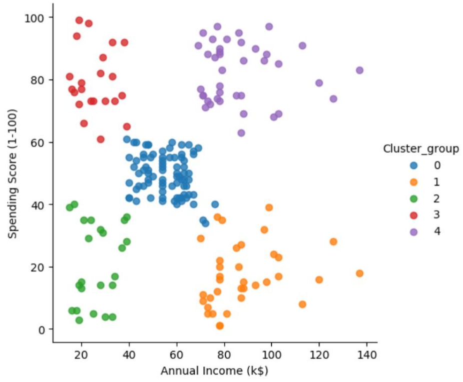
- Requires **K** to be predefined.
- Works well for **spherical, evenly distributed data**.

**Assignment Problem Statement**:

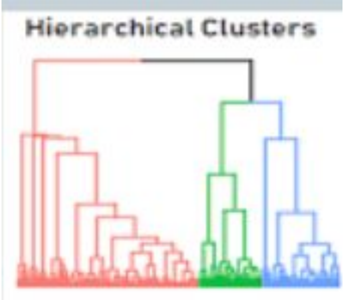
A new customer joins. Predict which group or category they belong to so marketing campaigns or SMS messages can be cu

**GitHub:** GeekPri/ML-Clustured-Algorithm-Unsupervised-

**Output:**

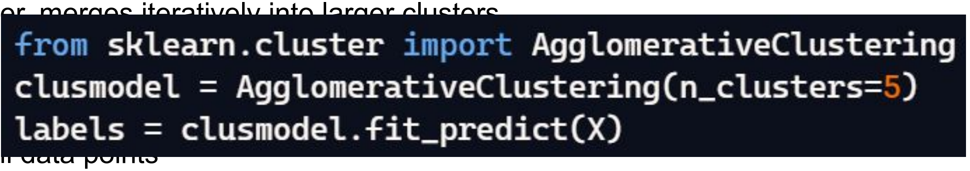


# Hierarchical Algorithm



## Type 1: Agglomerative

Starts with each point as its own cluster, merges iteratively into larger clusters.

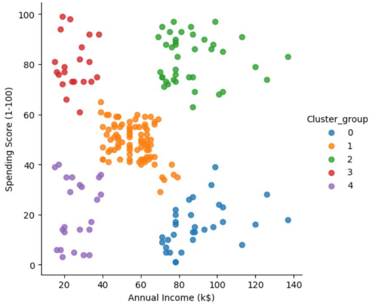


```
from sklearn.cluster import AgglomerativeClustering
clusmodel = AgglomerativeClustering(n_clusters=5)
labels = clusmodel.fit_predict(X)
```

## Type 2: Divise

Start with one big cluster containing all data points

Agglomerative Output:

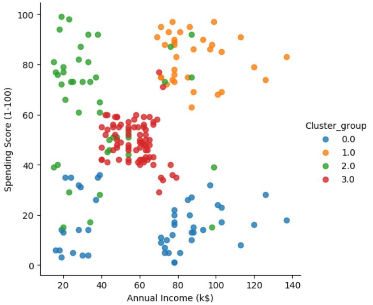


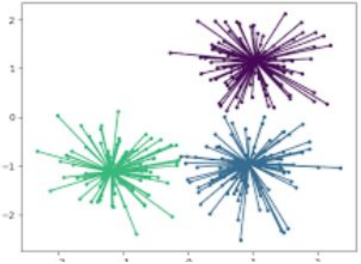
**Assignment Problem Statement**:

A new customer joins. Predict which group or category they belong to so marketing campaigns or SMS messages can be customer-focused.

**GitHub:** [ML-Clustured-Algorithm-Unsupervised-/Hierarchical -Agglomerative Clustering_ MallCustomers.ipynb at main · GeekPri/ML-Clustured-Algorithm-Unsupervised-]

Divisive Output:





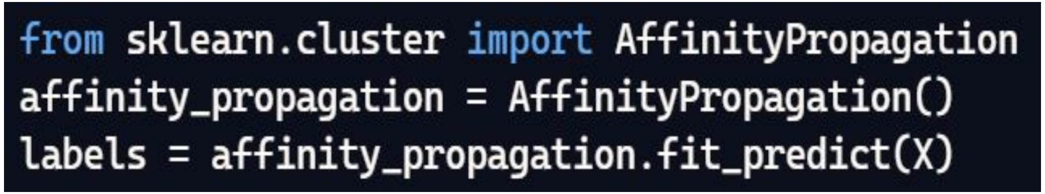
# Affinity Propagation - message passing

**How It Works**:

Data points "talk" to each other through iterative message passing.

**Key Features**:

Automatically determines the number of clusters by identifying exemplars (representative points).



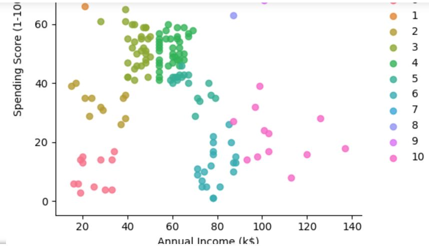
```
from sklearn.cluster import AffinityPropagation
affinity_propagation = AffinityPropagation()
labels = affinity_propagation.fit_predict(X)
```

**Assignment Problem Statement**:

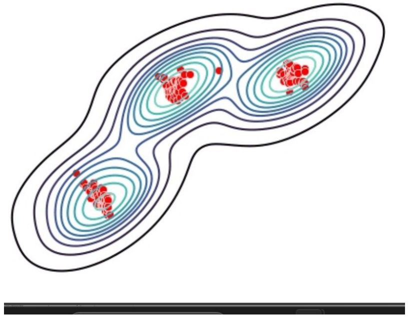
A new customer joins. Predict which group or category they belong to so marketing campaigns or SMS messages can be customer-focused.

**GitHub:** [Affinity Propagation]()

Output:

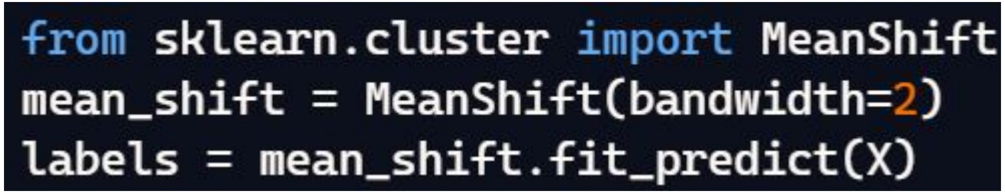


# Mean Shift



**Key Feature**:

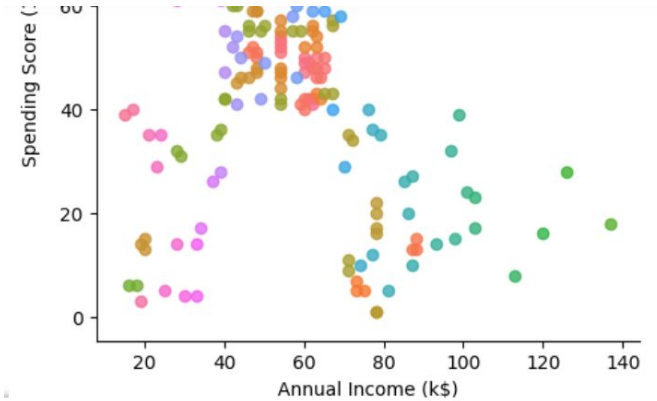
- No predefined number of clusters.
- Points move toward **densest regions** iteratively (bandwidth-controlled).



```
from sklearn.cluster import MeanShift
mean_shift = MeanShift(bandwidth=2)
labels = mean_shift.fit_predict(X)
```

GitHib:Mean Shift

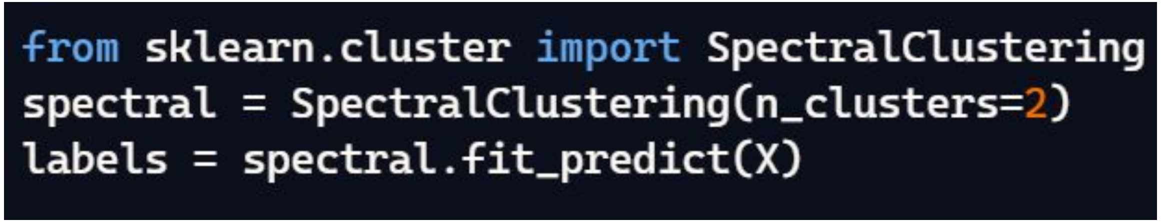
Output:



# Spectral Clustering

**Key Feature**:

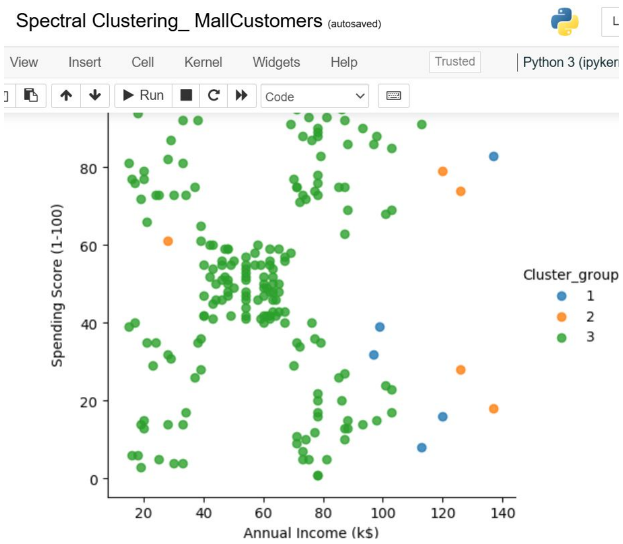
- Useful for **complex, non-linear data**.
- Treats points as nodes in a graph, clusters them by mapping to low-dimensional space.



```
from sklearn.cluster import SpectralClustering
spectral = SpectralClustering(n_clusters=2)
labels = spectral.fit_predict(X)
```

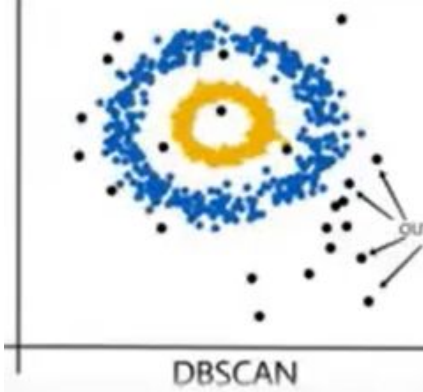
Github: Spectral Clustering

Output:



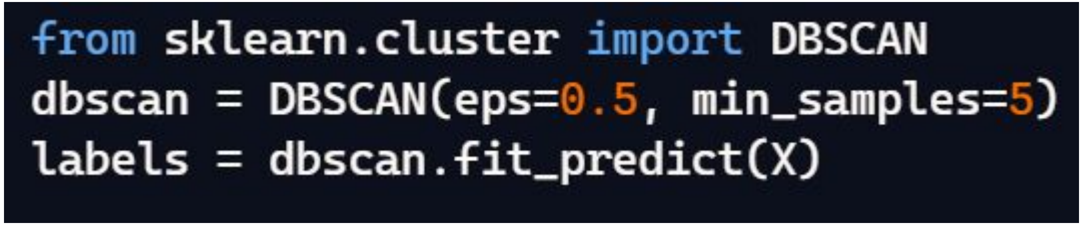
Spectral Clustering_ MallCustomers (autosaved)

# DBSCAN Density based



**Key Feature**:

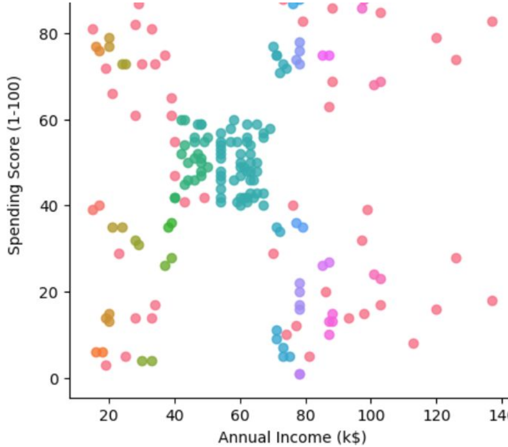
- Does not require the number of clusters.
- Groups **densely packed points**; labels sparse points as outliers.



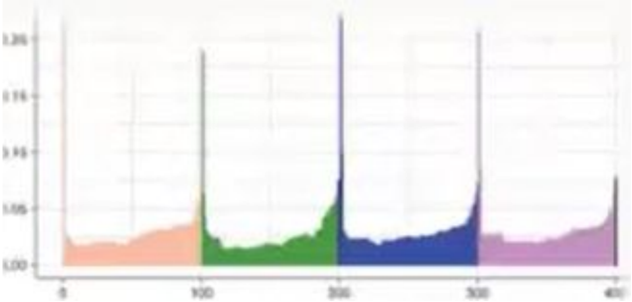
```
from sklearn.cluster import DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=5)
labels = dbscan.fit_predict(X)
```

Github: [DBScan]

Output

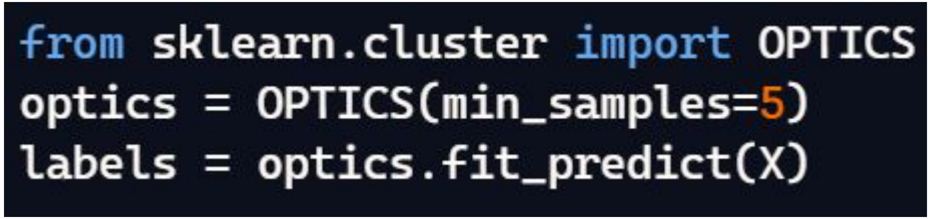


# OPTICS - Ordering the data points



**Key Feature**:

- Similar to DBSCAN but detects clusters of **varying densities**.
- Orders points to reflect density structure.

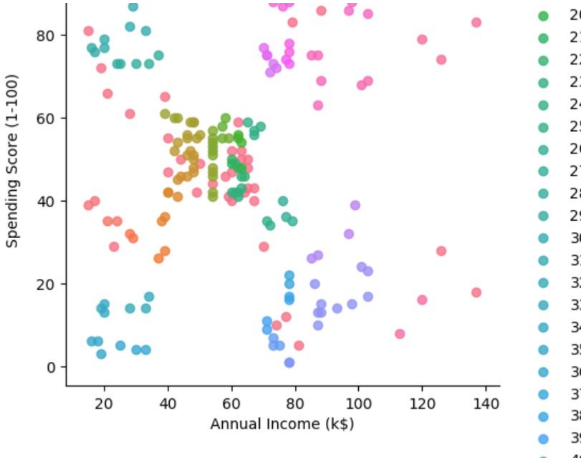


```
from sklearn.cluster import OPTICS
optics = OPTICS(min_samples=5)
labels = optics.fit_predict(X)
```

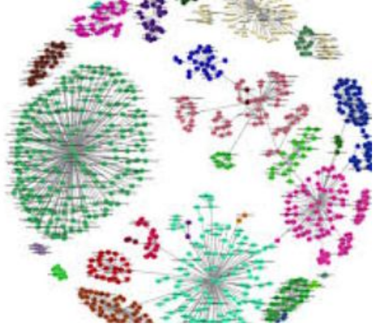
Github:

[ML-Clustured-Algorithm-Unsupervised-/Optics Clustering_ MallCustomers .ipynb at main · GeekPri/ML-Clustured-Algorithm-Unsupervised-]()

Output:

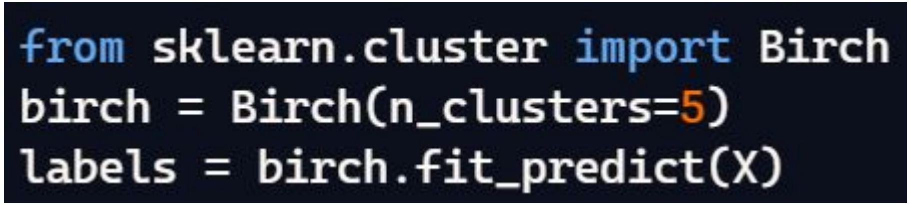


# Birch Algorithm



**Key Feature**:

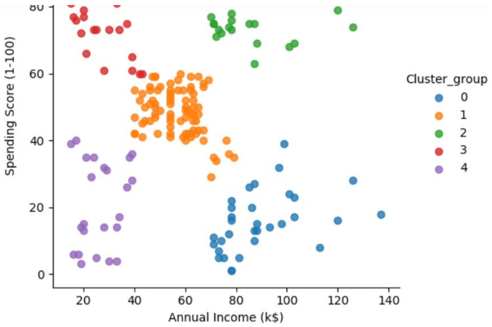
- Summarizes large datasets into smaller representations (CF Tree).
- Fast and memory-efficient.



```
from sklearn.cluster import Birch
birch = Birch(n_clusters=5)
labels = birch.fit_predict(X)
```

Githib: [Birch Algorithm]()

Output:



# Algorithm Comparison

| Algorithm | Class Name | Logic | Predefined Clusters | Density-Based | Scalable for Large Data |
|---|---|---|---|---|---|
| K-Means | KMeans | Groups data by finding **K centroids** | Yes | No | Medium |
| Hierarchical | AgglomerativeClustering | B**uilds a tree** by merging or splitting clusters | No | No | Low |
| Affinity Propagation | AffinityPropagation | **Passes messages** to find leaders (exemplars) | No | No | Low |
| Mean Shift | MeanShift | Shifts points to the **mean density region** | No | Yes | Low |
| Spectral Clustering | SpectralClustering | **Clusters using the spectrum** (graph) of data | Yes | No | Low |
| DBSCAN | DBSCAN | **Groups dense areas and labels** sparse points | No | Yes | Medium |
| OPTICS | OPTICS | **Orders points** to reflect density structure | No | Yes | Medium |
| BIRCH | Birch | Summarizes data into a **compact tree** | Yes | No | High |