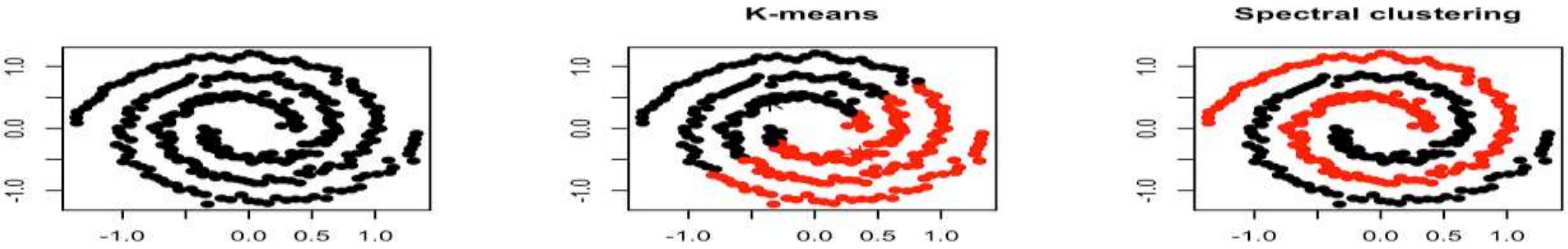
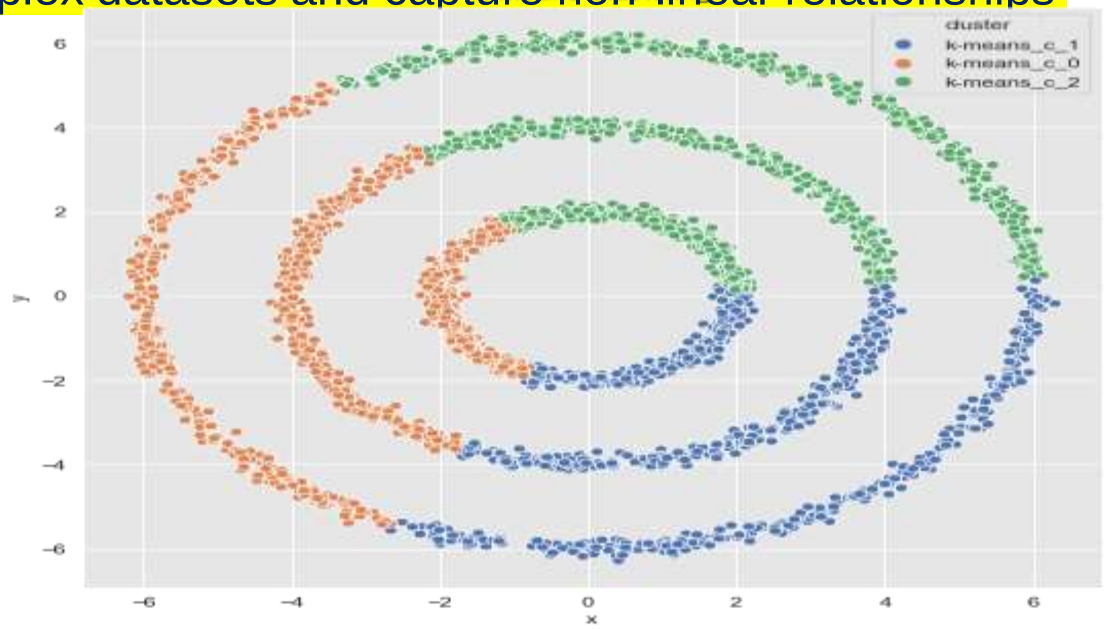
SPECTRAL CLUSTERING

- **Spectral clustering** is a powerful technique used for clustering data points based on their similarity. Unlike traditional clustering algorithms, spectral clustering doesn't make strong assumptions about the shape or size of the clusters. Instead, it leverages the eigenvalues and eigenvectors of a similarity matrix derived from the data to perform clustering.
- **Similarity Matrix Construction:**
- Spectral clustering begins by constructing a similarity matrix based on pairwise similarities between data points. Common measures include the radial basis function (RBF) kernel or nearest neighbors.
- **Dimensionality Reduction:**
- The eigenvectors corresponding to the smallest eigenvalues are used to embed the data into a lower-dimensional space. Typically, the first k eigenvectors are selected, where k represents the number of desired clusters.
- **Clustering in Reduced Space:**
- Traditional clustering algorithms (k-means) are applied to the reduced-dimensional space to partition the data into clusters.



SPECTRAL CLUSTERING

- **Advantages of Spectral Clustering:**
- **Flexibility**: Spectral clustering can detect clusters of arbitrary shapes and sizes.
- **Robustness**: It is robust to noise and outliers in the data.
- **Performance**: Spectral clustering often outperforms traditional methods, especially for complex datasets.
- **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.



SPECTRAL CLUSTERING

Here's a breakdown of the Python code, focusing on the application of Spectral Clustering

- **import pandas as pd**
- **import numpy as np**
- **import matplotlib.pyplot as plt**
- These lines import the necessary libraries: pandas for data manipulation, numpy for numerical operations.

- **dataset=pd.read_csv("Mall_Customers.csv")**

  This line reads the dataset from the CSV file named "Mall_Customers.csv" into a pandas DataFrame named dataset.
- **X=dataset.iloc[:,3:5].values**
- This line extracts the features (columns 3 and 4) from the dataset and assigns them to the variable X

SPECTRAL CLUSTERING

- from sklearn.cluster import SpectralClustering
- from sklearn.preprocessing import StandardScaler, normalize
- from sklearn.decomposition import PCA
- from sklearn.metrics import silhouette_score

- These lines import the Spectral Clustering class from scikit-learn, as well as other necessary modules for preprocessing and evaluation.

- spectral_model_rbf = SpectralClustering(n_clusters = 2, affinity ='rbf')
- labels_rbf = spectral_model_rbf.fit_predict(X_principal)

- This initializes the SpectralClustering model with 2 clusters using the radial basis function (RBF) kernel as the similarity measure and assigns cluster labels to the data.

SPECTRAL CLUSTERING

- plt.scatter(X_principal['P1'], X_principal['P2'],
- c = SpectralClustering(n_clusters = 2, affinity ='rbf').fit_predict(X_principal), cmap =plt.cm.winter)
- plt.show()

- This creates a scatter plot of the data points in the reduced 2-dimensional space, with colors representing different clusters obtained from spectral clustering with the RBF affinity.

- spectral_model_nn = SpectralClustering(n_clusters = 3, affinity ='nearest_neighbors')
- labels_nn = spectral_model_nn.fit_predict(X_principal)

- This initializes another Spectral Clustering model with 3 clusters using the nearest neighbors affinity and assigns cluster labels to the data.

SPECTRAL CLUSTERING

- plt.scatter(X_principal['P1'], X_principal['P2'],
- c = SpectralClustering(n_clusters = 3, affinity ='nearest_neighbors').fit_predict(X_principal), cmap =plt.cm.winter)
- plt.show()

- This creates a scatter plot of the data points, similar to the previous plot, but using spectral clustering with nearest neighbors affinity and 3 clusters.

- Overall, your code performs spectral clustering on the Mall Customers dataset, reduces the dimensionality of the data using PCA, and visualizes the clusters in a 2-dimensional space. It demonstrates how spectral clustering can be applied to group data points based on their similarities in reduced dimensions.

SPECTRAL CLUSTERING