**Clustering Analysis**

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

**Why is Clustering in Artificial Intelligence Important?**

The primary use of clustering in machine learning is to extract valuable inferences from many unstructured data sets. If you are working with large amounts of data that are also not structured, it is only logical to organize that data to make it helpful in so many other ways, and clustering helps us do that. Clustering and classification allow you to take a sweeping glance at your data. And then form some logical structures based on what you find there before going deeper into the nuts-and-bolts analysis. Clustering is a significant component of machine learning, and its importance is highly significant in providing better machine learning techniques.

**Benefits for business**

**Marketing:**

In the field of marketing, clustering can be used to identify various customer groups with existing customer data. Based on that, customers can be provided with discounts, offers, promo codes etc.

**Real Estate:**

Clustering can be used to understand and divide various property locations based on value and importance. Clustering algorithms can process through the data and identify various groups of property on the basis of probable price.

**BookStore and Library management:**

Libraries and Bookstores can use Clustering to better manage the book database. With proper book ordering, better operations can be implemented.

**Document Analysis:**

Often, we need to group together various research texts and documents according to similarity. And in such cases, we don’t have any labels. Manually labelling large amounts of data is also not possible. Using clustering, the algorithm can process the text and group it into different themes.

**Example:**

This example helps to understand the customers like who can be easily converge [Target Customers] so that the sense can be given to marketing team and plan the strategy accordingly.

CELL1:

# Importing essential libraries

import numpy as np

import pandas as pd

# Loading the dataset

df = pd.read\_csv("Mall\_Customers.csv")

CELL2:

# Returns number of rows and columns of the dataset

df.shape

# Returns an object with all of the column headers

df.columns

# Returns different datatypes for each columns (float, int, string, bool, etc.)

df.dtypes

# Returns the first x number of rows when head(x). Without a number it returns 5

df.head()

# Returns the last x number of rows when tail(x). Without a number it returns 5

df.tail()

# Returns basic information on all columns

df.info()

# Returns basic statistics on numeric columns

df.describe().T

# Returns true for a column having null values, else false

df.isnull().any()

CELL3:

# Creating the copy of dataset

df\_copy = df.copy(deep=True)

df\_copy.head(3)

# Dropping the column of 'CustomerID' as it does not provide any value

df\_copy.drop('CustomerID', axis=1, inplace=True)

df\_copy.columns

CELL4:

# Loading essential libraries

import matplotlib.pyplot as plt

import seaborn as sns

df\_copy.columns

CELL5:

# Visualising the columns 'Gender' using Countplot

sns.countplot(x='Gender', data=df\_copy)

plt.xlabel('Gender')

plt.ylabel('Count')

CELL6:

# Visualising the columns 'Age' using Histogram

plt.hist(x=df\_copy['Age'], bins=10, orientation='vertical', color='red')

plt.xlabel('Age')

plt.ylabel('Count')

plt.show()

CELL7:

# Visualising the columns 'Age', 'Spending Score (1-100)' using Scatterplot and Jointplot

sns.scatterplot(data=df\_copy, x='Age', y='Spending Score (1-100)', hue='Gender')

sns.jointplot(data=df\_copy, x='Age', y='Spending Score (1-100)')

CELL8:

# Visualising the columns 'Annual Income (k$)', 'Spending Score (1-100)' using Scatterplot and Jointplot

sns.scatterplot(data=df\_copy, x='Annual Income (k$)', y='Spending Score (1-100)', hue='Gender')

sns.jointplot(data=df\_copy, x='Annual Income (k$)', y='Spending Score (1-100)')

CELL9:

# Selecting 'Annual Income' and 'Spending Score' as the features for clustering

X = df\_copy.iloc[:, [2,3]]

X.columns

CELL10:

# Calculating WCSS values for 1 to 10 clusters

from sklearn.cluster import KMeans

wcss = []

for i in range(1,11):

  kmeans\_model = KMeans(n\_clusters=i, init='k-means++', random\_state=42)

  kmeans\_model.fit(X)

  wcss.append(kmeans\_model.inertia\_)

  # Plotting the WCSS values

plt.plot(range(1,11), wcss)

plt.title('Elbow Method')

plt.xlabel('Number of Clusters')

plt.ylabel('WCSS')

plt.show()

CELL11:

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X = scaler.fit\_transform(X)

# Training the KMeans model with n\_clusters=5

kmeans\_model = KMeans(n\_clusters=5, init='k-means++', random\_state=42)

y\_kmeans = kmeans\_model.fit\_predict(X)

# Visualising the clusters

plt.scatter(X[y\_kmeans == 0, 0], X[y\_kmeans == 0, 1], s = 30, c = 'yellow', label = 'Cluster 1')

plt.scatter(X[y\_kmeans == 1, 0], X[y\_kmeans == 1, 1], s = 30, c = 'cyan', label = 'Cluster 2')

plt.scatter(X[y\_kmeans == 2, 0], X[y\_kmeans == 2, 1], s = 30, c = 'lightgreen', label = 'Cluster 3')

plt.scatter(X[y\_kmeans == 3, 0], X[y\_kmeans == 3, 1], s = 30, c = 'orange', label = 'Cluster 4')

plt.scatter(X[y\_kmeans == 4, 0], X[y\_kmeans == 4, 1], s = 30, c = 'red', label = 'Cluster 5')

plt.scatter(x=kmeans\_model.cluster\_centers\_[:, 0], y=kmeans\_model.cluster\_centers\_[:, 1], s=100, c='black', marker='+', label='Cluster Centers')

plt.legend()

plt.title('Clusters of customers')

plt.xlabel('Annual Income')

plt.ylabel('Spending Score')

plt.show()