**3-topshiriq.**

**Mavzu:** Mashinali o‘qitishda o‘qituvchisiz o‘qitish algoritmlari yordamida klasterlash masalalarini yechish.

**Ishdan maqsad:** Talabalarda o‘qituvchisiz o‘qitish (unsupervised learning) algoritmlari haqida tushuncha paydo qilish va klasterlash masalalari ishlab chiqish ko‘nikmalarini shakllantirish.

**Ishning nazariy asoslari**

Mashinali o‘qitishda o‘qituvchisiz o‘qitish algoritmlari ma’lumotlarni to‘g‘ridan-to‘g‘ri belgilarsiz tahlil qilish va klasterlashga asoslanadi. O‘qituvchisiz o‘qitish algoritmlarida ma’lumotlar o‘quv jarayonida maxsus belgilar yoki guruhlar bo‘lmagan holatda ishlatiladi.

Klasterlash masalalarni yechishda quyidagi algoritmlardan foydalanish mumkin:

1. K-Means klasterlash: Ma’lumotlar nuqtalari oldindan belgilangan klasterlar soniga (K) asoslanib, bir nechta klasterlarga bo‘linadi. Har bir klasterning markazi hisoblanadi va ma’lumot nuqtalari eng yaqin markazga qarab klasterlanadi.
2. Iyerarxik klasterlash: Bu algoritm ma’lumotlarni iyerarxik tarzda klasterlaydi, ya’ni avval kichik klasterlar yaratiladi, so‘ng ular yirikroq klasterlarga birlashtiriladi.
3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Ma’lumotlarni zichlik asosida klasterlarga ajratadi va izolyatsiya qilingan ma’lumot nuqtalarini ajratadi.

**Ishni bajarish tartibi**

Python dasturlash tilida K-means klasterlash algoritmidan foydalanib ma’lumotlarni klasterlanadi.

***1-qadam. Kerakli kutubxonalarni o‘rnatib olish.***

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.datasets import make\_blobs

***2-qadam. Sun’iy ma’lumotlarni ishlab chiqish yoki mavjud dateset ni ulab olish.***

X, y = make\_blobs(n\_samples=300, centers=3, random\_state=42, cluster\_std=0.60)

plt.scatter(X[:, 0], X[:, 1], s=50)

plt.show()

***3-qadam. K-Means algoritmi yordamida sun’iy ma’lumotlarni klasterlarga ajratiladi.***

kmeans = KMeans(n\_clusters=3)

kmeans.fit(X)

centroids = kmeans.cluster\_centers\_

labels = kmeans.labels\_

plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')

plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200, alpha=0.75)

plt.show()

***4-qadam. K-Means algoritmi natijalarini tahlil qilish va natijalarni yaxshilash.***

K-Means algoritmi ma’lumotlarni belgilangan 3 ta klasterga ajratadi va har bir klasterning markazlarini hisoblaydi. Grafikda har bir ma’lumot nuqtasi o‘z klasteriga mos rangda ko‘rsatiladi, klaster markazlari esa qizil nuqta sifatida tasvirlanadi.

K-Means dastlab klasterlar sonini oldindan belgilashni talab qiladi. Agar aniq klasterlar sonini bilinmasa, “Elbow Method” yordamida optimal klasterlar sonini aniqlash mumkin.

inertia\_values = []

for k in range(1, 10):

kmeans = KMeans(n\_clusters=k)

kmeans.fit(X)

inertia\_values.append(kmeans.inertia\_)

plt.plot(range(1, 10), inertia\_values, 'bo-')

plt.xlabel('Klasterlar soni')

plt.ylabel('Inertia')

plt.show()

Boshqa klasterlash algoritmlaridan biri DBSCAN yordamida klasterlashni ham ko‘rish mumkin. DBSCAN klasterlash usulida zichlikka asoslangan klasterlar ajratiladi.

from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

dbscan.fit(X)

plt.scatter(X[:, 0], X[:, 1], c=dbscan.labels\_, cmap='plasma')

plt.title("DBSCAN Klasterlash")

plt.show()

**Xulosa.** K-Means va DBSCAN kabi o‘qituvchisiz o‘qitish algoritmlari ma’lumotlarni avtomatik ravishda klasterlarga ajratadi. Elbow Method yordamida K-Means algoritmasida optimal klasterlar sonini aniqlash mumkin.

 



