**DBSCAN**

**(Density-Based Spatial Clustering of Aplication with Noise)**

* DBSCAN utilizează densitatea punctelor ca factor principal în crearea clusterelor
* DBSCAN permite crearea clusterelor dar și detecția valorilor aberante (outliers) prin setarea a 2 parametri: epsilon și numărul minim de puncte
* DBSCAN clasifică toate datele în puncte de tip: nucleu (core), limită (border) și valoare aberantă (outlier)
* Un punct devine core dacă în raza sa de căutare specificată de parametrul epsilon se află un număr dat de puncte (inclusiv acest punct) specificat de parametrul numărul minim de puncte.
* Un punct devine border dacă în raza sa de căutare specificată de parametrul epsilon se află un număr mai mic de puncte (inclusiv acest punct) decât valoarea parametrului numărul minim de puncte.
* Un punct devine outlier dacă acesta nu se include în raza unui alt punct și nici în raza sa de căutare nu se află nici un alt punct.
* Punctele core permit crearea unui cluster sau includerea unui cluster existent, punctele border doar se includ într-un cluster existent, iar punctele outlier nu formează și nici nu se includ în nici un cluster
* Epsilon – raza din jurul unui punct unde are loc căutarea numărului minim de puncte pentru a determina tipul acestui punct.
* Numărul minim de puncte – este numărul minim de puncte (inclusiv și punctul curent) din raza Epsilon a unui punct care se va lua în considerare atunci când se va determina tipul acestui punct
* Etapele algoritmului DBSCAN:
* Se selectează aliator un punct care nu se include în nici un cluster
* Se determină tipul acestui punct în funcție de parametrii epsilon și numărul minim de puncte
* Dacă punctul este de tip core atunci se formează un cluster în care se vor include toate punctele din raza sa de căutare și se verifică tipul acestora.
* Dacă tipul unui alt punct din cluster este core atunci în clusterul dat se includ și punctele din raza sa de căutare
* Dacă tipul unui punct este border atunci acest punct se va considera punct limită a clusterului iar punctele din raza lui de căutare nu se includ în cluster
* Dacă punctul este de tip outlier atunci nu se creează nic un cluster
* Vizualizarea online a formării clusterelor DBSCAN <https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/>
* Distanța Epsilon și numărul minim de puncte permit setarea automată a numărului de clustere și a numărului de outliers
* Pentru determinarea valorilor optime a parametrilor Epsilon și numarul minim de puncte se vor evalua dependențele de acești parametri a unuia dintre valorile: numărul de clustere, numărul de outlier, sau procentajul outliers
* Algoritmul de elaborare a modelului DBSCAN și compararea cu K-Means Clustering în Scikit-Learn:
* importul a trei seturi de date
* vizualizarea grafica a datelor df1 ***sns.scatterplot(data=df1,x='X1',y='X2')***
* elaborarea unei functii de realizarea a predictiei si afisare a clusterelor ***afisare\_clustere(model,data)***
* importul modulului KMeans from ***sklearn.cluster import KMeans***
* crearea modelului KMeans pentru 3 clustere ***model\_K\_3 = KMeans(n\_clusters = 3)***
* apelarea funcției de realizarea predictie pe datele df1 ***afisare\_clustere(model\_K\_3, df1)***
* # crearea modelului KMeans pentru 2 clustere ***model\_K\_2 = KMeans(n\_clusters = 2)***
* apelarea funcției de realizarea predictie pe datele df2 ***afisare\_clustere(model\_K\_2, df2)***
* importul modulului DBSCAN ***from sklearn.cluster import DBSCAN***
* crearea modelului DBSCAN cu parametrul epsilon =0,6 ***model\_D\_1 = DBSCAN(eps=0.6)***
* apelarea functiei de realizarea predictie pe datele df1 ***afisare\_clustere(model\_D\_1, df1)***
* crearea modelului DBSCAN cu paramatrul epsilon =0,15 ***model\_D\_2 = DBSCAN(eps=0.15)***
* apelarea functiei de realizarea predictie pe datele df2 ***afisare\_clustere(model\_D\_2, df2)***
* Selectarea valorilor hiperparametrilor modelului DBSCAN în Scikit-Learn:
* vizualizarea hiperparametrilor impliciti ai modelului DBSCAN ***model\_D.get\_params(deep=True)***
* crearea modelului DBSCAN cu hiperparametrul epsilon de valoarea foarete mica ***dbscan = DBSCAN(eps=0.001)***
* apelarea functiei de realizarea predictie pe datele df5 ***afisare\_clustere(dbscan, df5)***
* crearea modelului DBSCAN cu hiperparametrul epsilon de valoarea foarte mare ***dbscan = DBSCAN(eps=10)***
* apelarea functiei de realizarea predictie pe datele df5 ***afisare\_clustere(dbscan, df5)***
* crearea modelului DBSCAN cu hiperparametrul epsilon de o valoare acceptabila ***dbscan = DBSCAN(eps=1)***
* apelarea functiei de realizarea predictie pe datele df5 ***afisare\_clustere(dbscan, df5)***
* vizualizarea clusterelor datelor ***dbscan.labels\_***
* determinarea numarului de outliers ***np.sum(dbscan.labels\_ == -1)***
* crearea unei bucle de determinare a numarului de outliers pentru diferite valori a lui epsilon ***numar\_outliers***
* afisarea dependentei numarului de outliers de valoarea lui epsilon ***sns.lineplot(x=np.linspace(0.001,10,100),y=numar\_outliers)***
* crearea modelului DBSCAN cu hiperparametrul epsilon de o valoare optima ***dbscan = DBSCAN(eps=0.7)***
* apelarea functiei de realizarea predictie pe datele df5 ***afisare\_clustere(dbscan, df5)***
* crearea unei bucle de determinare a procentului de outliers pentru diferite valori a lui epsilon ***procent\_outliers***
* afisarea dependentei procentului de outliers de valoarea lui epsilon ***sns.lineplot(x=np.linspace(0.001,10,100),y=procent\_outliers)***
* crearea modelului DBSCAN cu hiperparametrul min\_samples de valoarea minima ***dbscan = DBSCAN(min\_samples=1)***
* apelarea functiei de realizarea predictie pe datele df5 ***afisare\_clustere(dbscan, df5)***
* crearea modelului DBSCAN cu setarea parametrilor eps si min\_samples ***dbscan = DBSCAN(eps=0.7, min\_samples=1)***
* apelarea functiei de realizarea predictie pe datele df5 ***afisare\_clustere(dbscan, df5)***
* crearea unei bucle de determinare a procentajului de outliers pentru diferite valori a numarului mimin ***procent\_outliers***
* afisarea dependentei procentajului de outliers de valoarea lui min\_samples ***sns.lineplot(x=np.arange(1,100),y=procent\_outliers)***
* crearea modelului DBSCAN cu hiperparametrul min\_samples de valoarea dubla numarului de caracteristici ale datelor ***dbscan = DBSCAN(min\_samples=2\*df5.shape[1])***
* apelarea functiei de realizarea predictie pe datele df5 ***afisare\_clustere(dbscan, df5)***