BIG DATA

CURS 9

Concepte și metode fundamentale ale analizei datelor

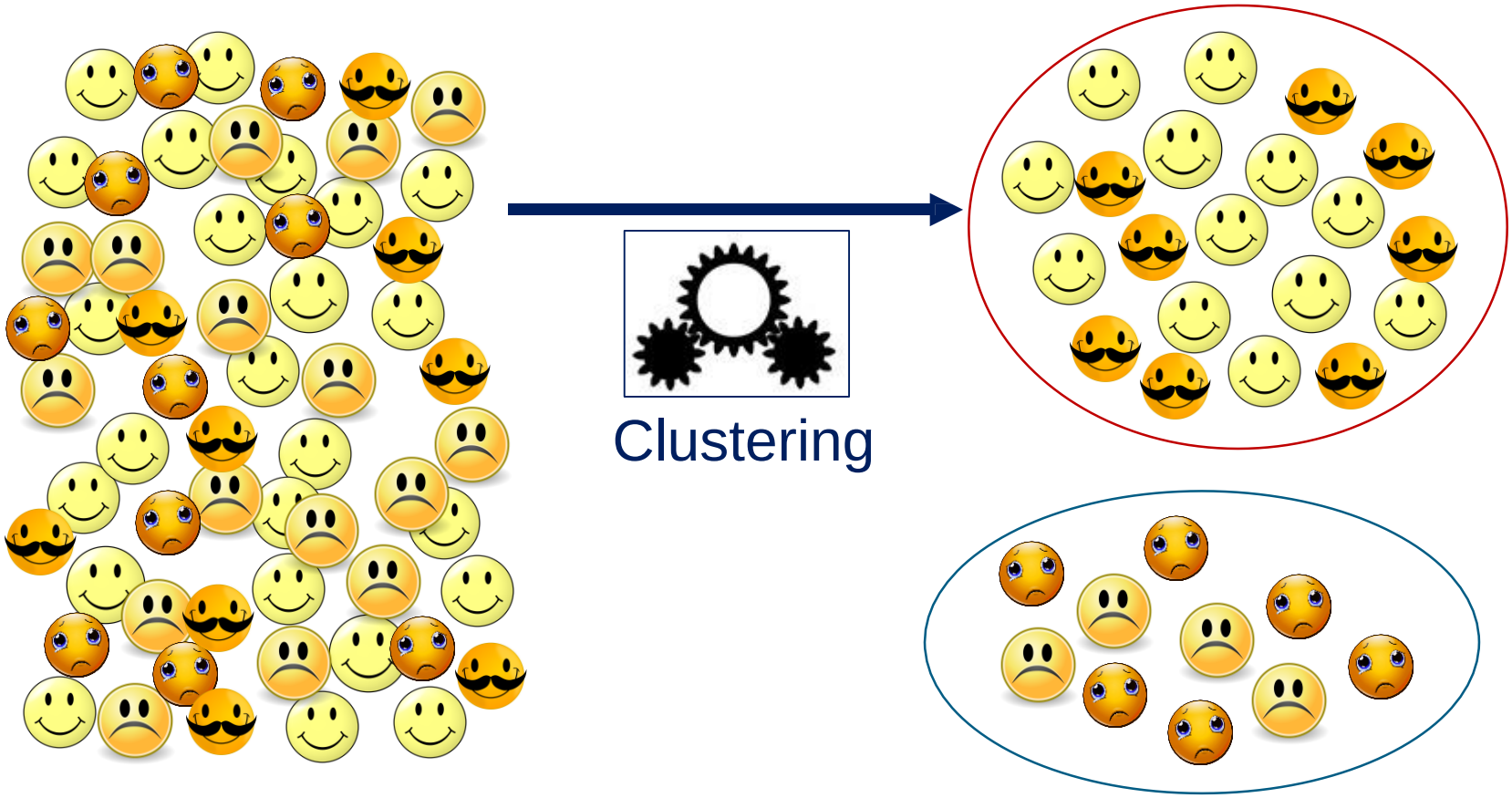
1. Etapele unui proiect Data Science
2. Explorarea datelor
3. Analiza datelor
4. Descoperirea regulilor de asociere
5. Clustering
6. Clasificare
7. Regresie

5. Clustering

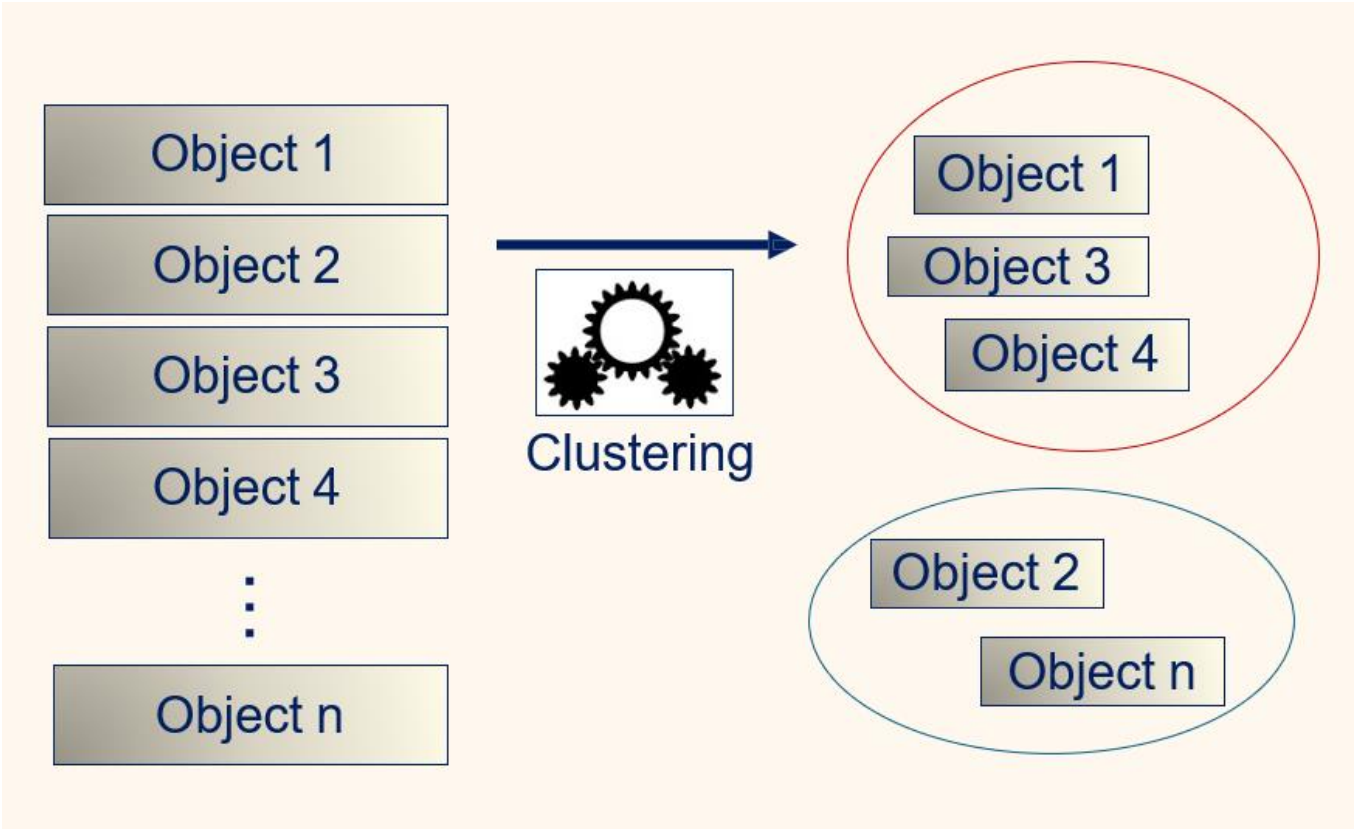
Clustering

- Descriere
- Algoritmi de Clustering:
  - K-Means
  - EM
  - DBSCAN
  - Single Linkage
- Comparaţie algoritmi clustering
- Concluzii

Clustering - exemplu



Problema generală



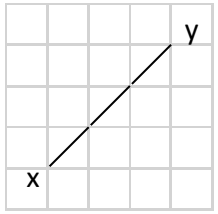
Problema formală

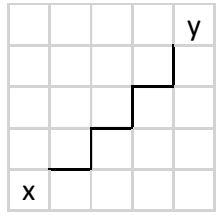
- Spațiu de obiecte
  - $O = \{object_1, object_2, \dots\}$
  - Poate fi infinit
- Reprezentări ale obiectelor într-un spațiu de caracteristici (numeric)
  - $\mathcal{F} = \{\phi(o)\ ,\ o \in O\}$
- Clustering
  - Gruparea obiectelor
  - Obiectele din același grup $g \in G$ trebuie să fie similare
  - $c: \mathcal{F} \rightarrow G$

Cum se poate măsura similaritatea?

Măsurarea distanțelor

- Similaritate -> distanță
  - Problema măsurării similarității se transformă într-o problemă de măsurare de distanțe
  - Distanța mică = similaritate
- Distanța euclidiană:
  - Se bazează pe norma euclidiană $||x||_2$
  - $d(x, y) = ||y - x||_2 = \sqrt{(y_1 - x_1)^2 + \cdots + (y_n - x_n)^2}$
- Distanța Manhattan
  - Se bazează pe norma Manhattan $||x||_1$
  - $d(x, y) = ||y - x||_1 = |y_1 - x_1| + \ldots + |y_n - x_n|$
- Distanța Cebîșev:
  - Se bazează pe norma maximă $\|x\|_\infty$
  - $d(x, y) = ||y - x||_\infty = \max_{i=1\ldots} |y_i - x_i|$





2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Evaluarea rezultatelor clustering-ului

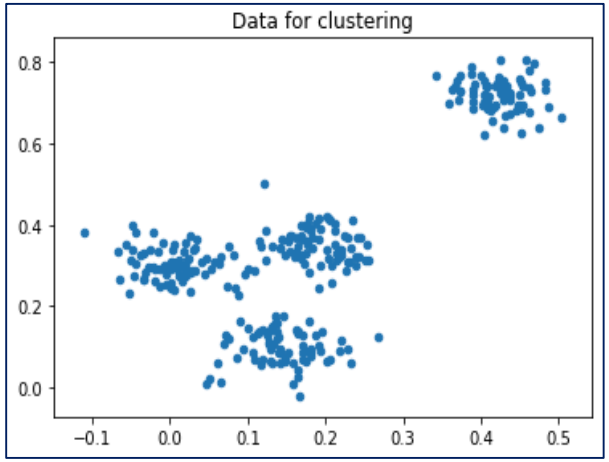
- Nu există metrici generale, depind de algoritmi
  - Varianța mică pentru *k*-means
  - Densitate mare pentru DBSCAN
  - Potrivire adecvată față de variabilele modelului pentru EM
- Frecvent, au loc verificări manuale
  - Au sens clusterele?
  - Poate fi dificilă o astfel de verificare deoarece:
    - Datele sunt de dimensiuni foarte mari
    - Pot fi multe clustere
    - Datele pot avea un număr mare de dimensiuni

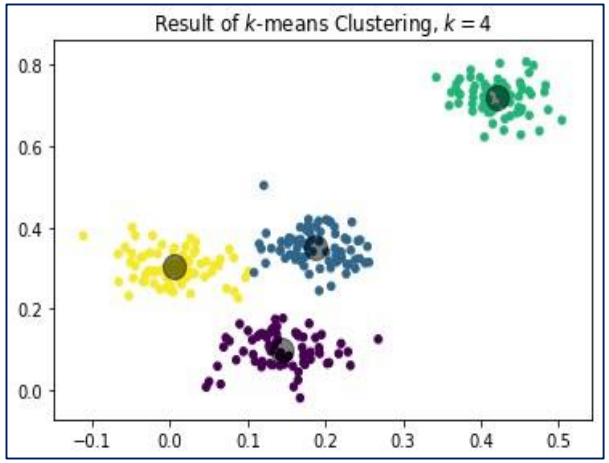
Algoritmi de clustering – *k-means*

- Clusterele sunt descrise prin intermediul centrelor lor
  - Un centru este denumit centroid
  - Clustering bazat pe centroizi

Cum se obțin centroizii?

- Obiectele sunt asignate celui mai apropiat centroid

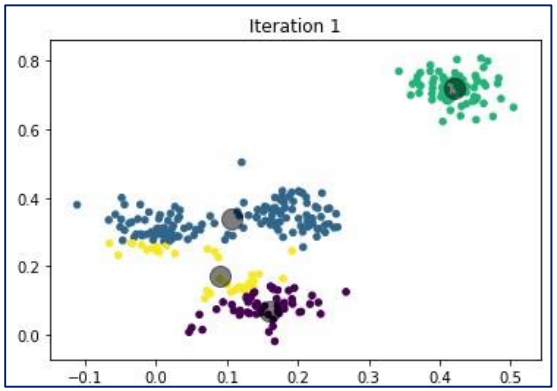


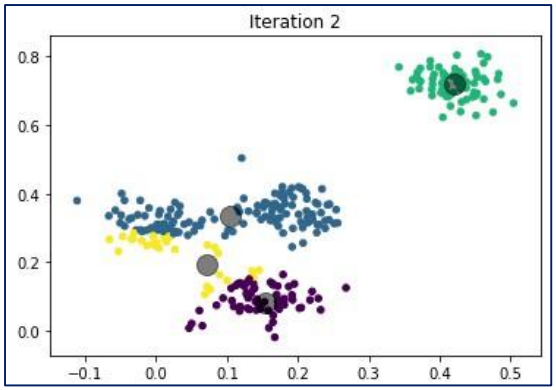


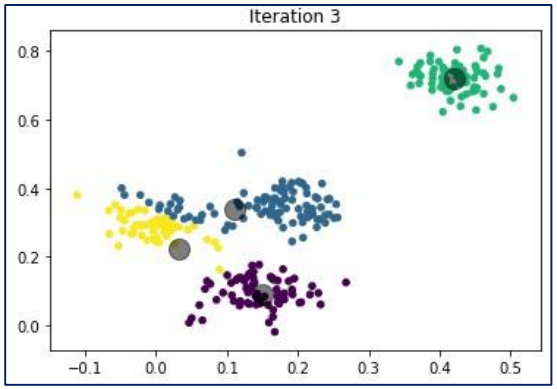
Algoritmul *k-means*

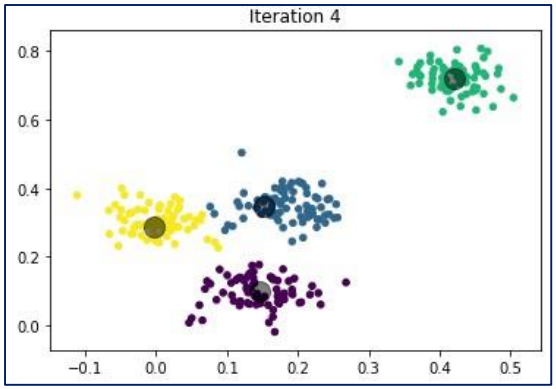
- Se selectează centroizii inițiali $C_1, \ldots, C_k$
  - Alegerea poate fi aleatoare
- Se asignează fiecare obiect celui mai apropiat centroid
  - $c(x) = \underset{i=1,\ldots,k}{\operatorname{argmin}} d(x, C_i)$
- Se actualizează centroidul
  - Media aritmetică a obiectelor asignate
  - $C = \frac{1}{|\{x \mid c(x)=i\}|} \sum_{x \mid c(x)=i} x_i$
- Se repetă pașii de acualizare și asignare
  - Până când are loc convergența sau
  - Până când se ajunge la numărul maxim de iterații

Vizualizarea algoritmului k-means





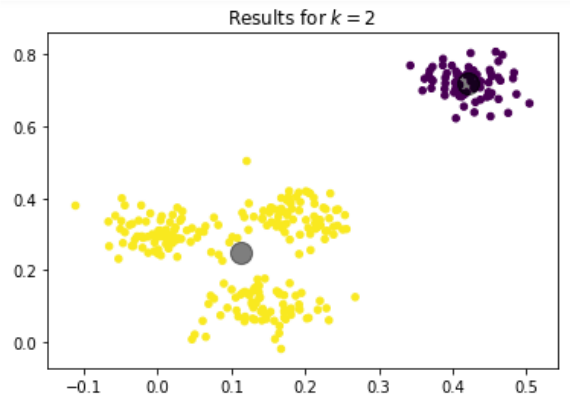


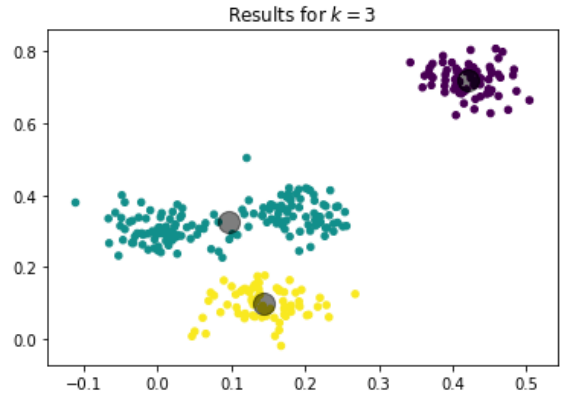


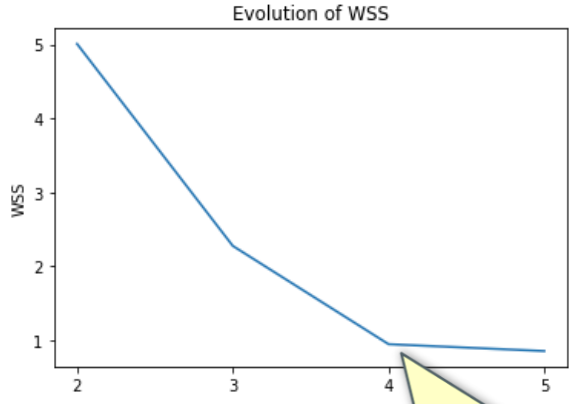
Selectarea lui k

- Pe baza intuiției și a cunoștințelor despre date
  - Pe baza vizualizărilor grafice
  - Pe baza cunoștințelor despre domeniu
- Având în vedere scopul
  - Număr fixat de grupări dorite
- Pe baza celei mai bune potriviri (*best fit*)
  - Within-Sum-of-Squares
  - $WSS = \sum_{i=1}^{k} \sum_{x \mid c(x)=i} d(x, C_i)^2$

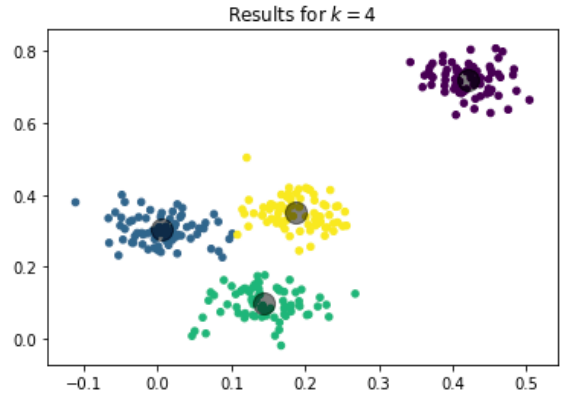
Rezultate pentru k=2, …,5

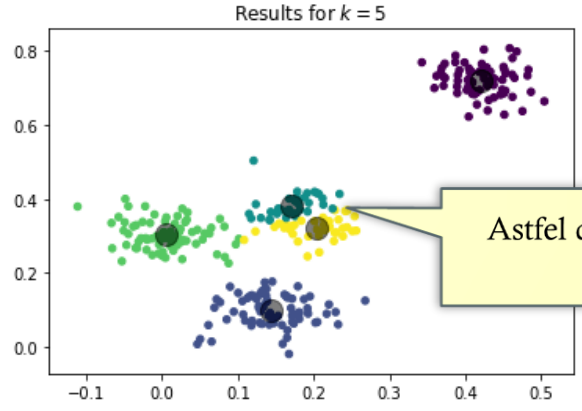






Modificările mari asupra pantei indică valori potențial bune pentru k



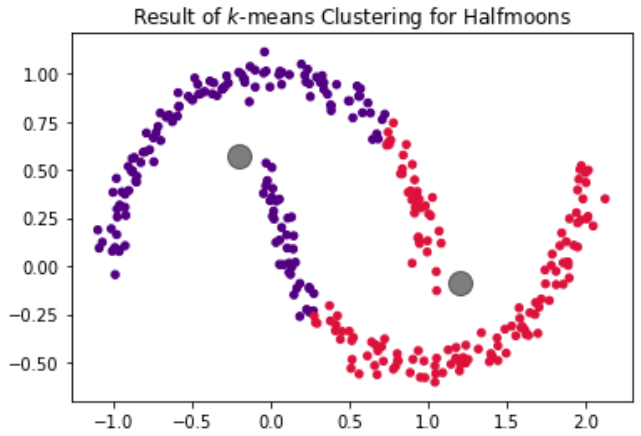


Astfel de diviziuni indică prea multe clustere

k = 2, 3, 4 sunt ok
Se folosesc cunoștințele despre domeniu pentru a decide

Probleme ale algoritmului *k-means*

- Depinde de cluster-ele inițiale
  - Rezultatele pot fi instabile
- Alegerea greșită a lui *k* poate conduce la rezultate greșite
- Toate caracteristicile trebuie să aibă o scală similară
  - Diferențele de scală introduc ponderi artificiale între caracteristici
  - Scalele mari le domină pe cele mici
- Funcționează bine pe clustere „rotunde”

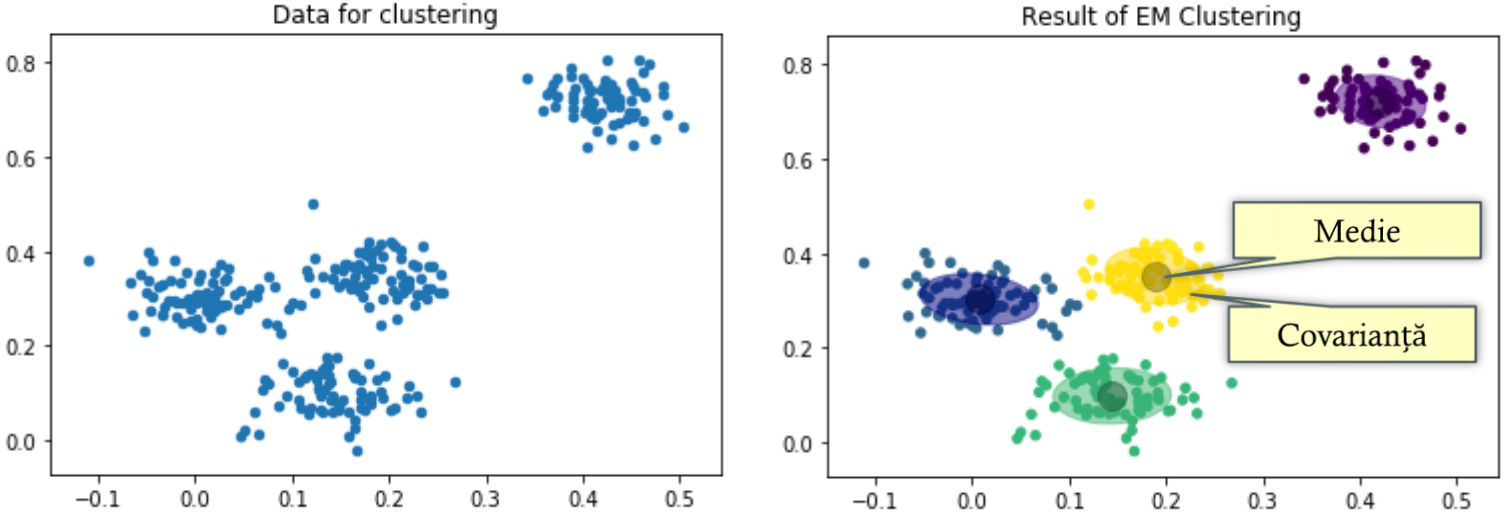


Clustering EM (Expectation-Maximization)

- Clusterele sunt descrise prin repartiții (distribuții) de probabilitate
  - De obicei, repartiția normală (*Gaussian Mixture Model*)
  - Clustering bazat pe repartiție

Cum obținem repartițiile?

- Obiectele sunt asignate clusterului „celui mai probabil”

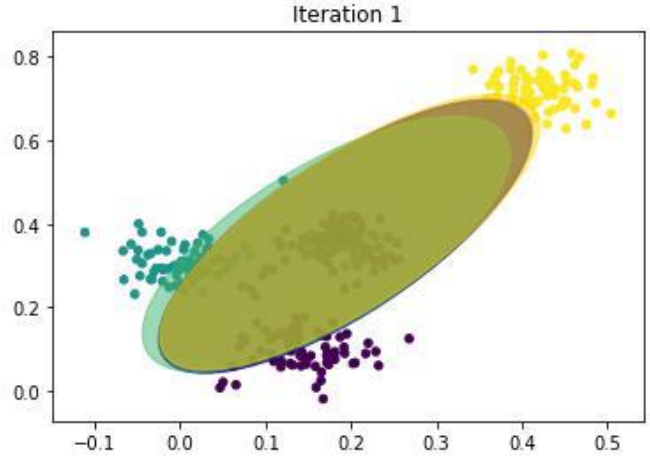


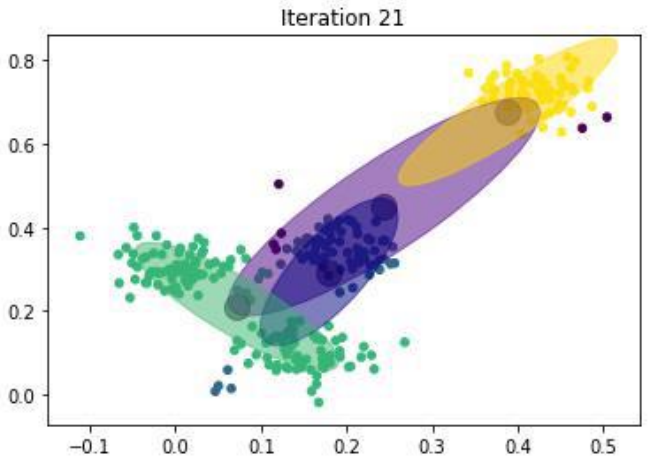
Algoritmul EM simplificat

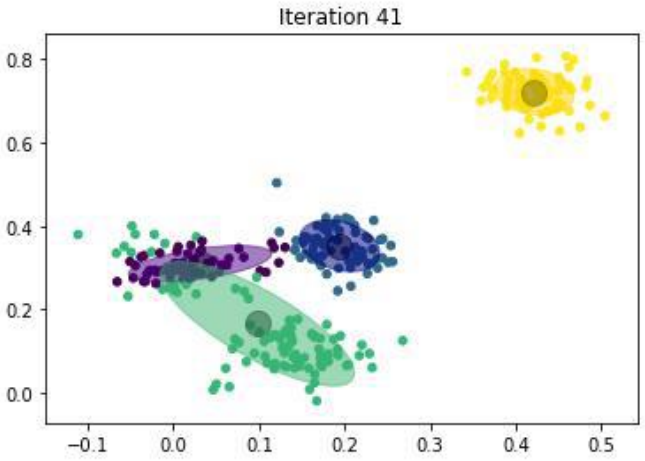
- Obiectiv: Determinarea a k repartiţii normale care încadrează (*fit*) datele.
  - $C_1 \sim (\mu_1, \sigma_1), \ldots, C_k \sim (\mu_k, \sigma_k)$
  - Valorile iniţiale se estimează asemănător metodei *k-means*
- Pasul de aşteptare (***Expectation***)
  - Se calculează ponderile obiectelor
  - Ponderile definesc probabilitatea ca un obiect să aparţină unui cluster
  - $w_j(x) = \frac{p(x|\mu_j,\sigma_j)}{\sum_{i=1}^{k} p(x|\mu_i,\sigma_i)}$ pentru toate obiectele $x \in X$
- Pasul de maximizare (***Maximization***)
  - Se actualizează valorile mediilor
  - $\mu_j = \frac{1}{|X|} \sum_{x \in X} w_j(x) \cdot x$

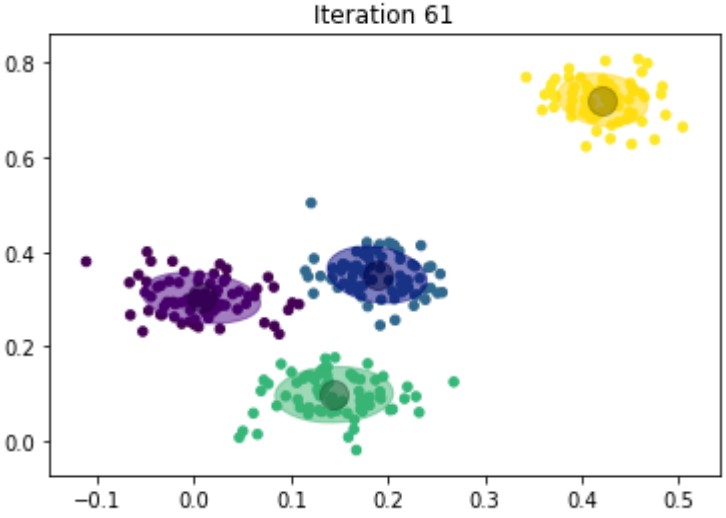
O versiune corectă, dar simplificată a algoritmului, ce ignoră actualizarea covarianţei.

Vizualizarea algoritmului EM





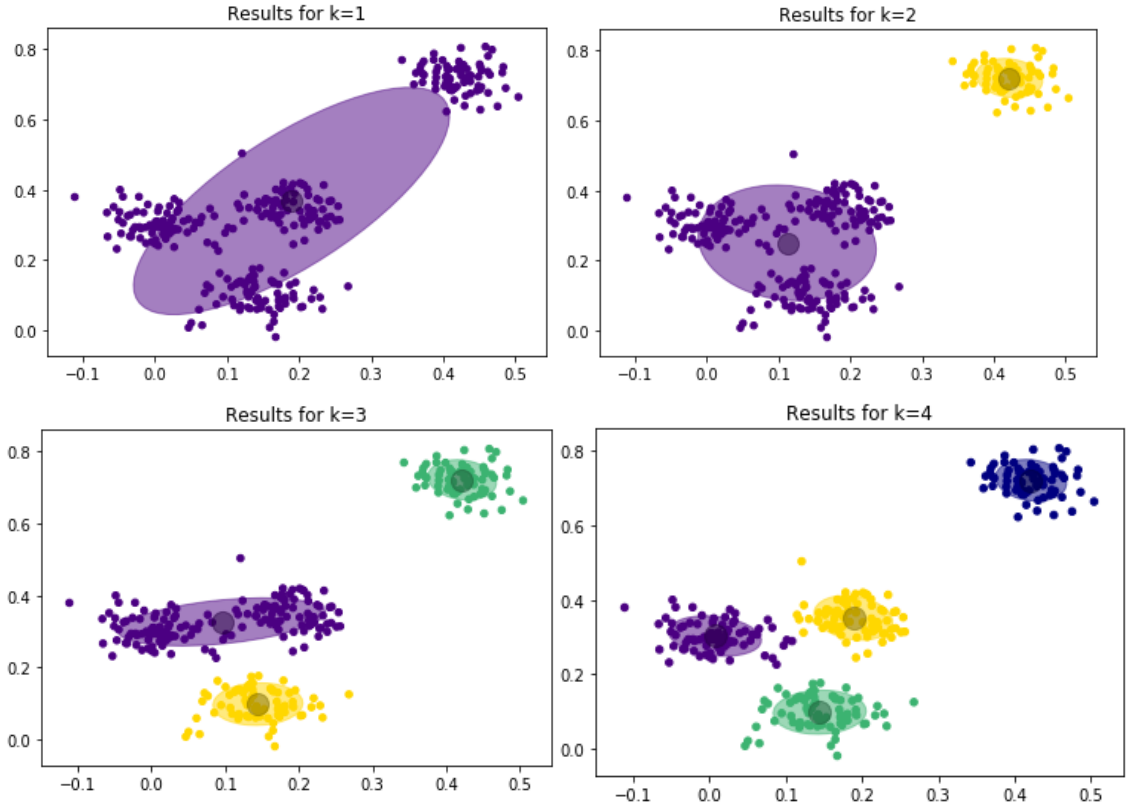


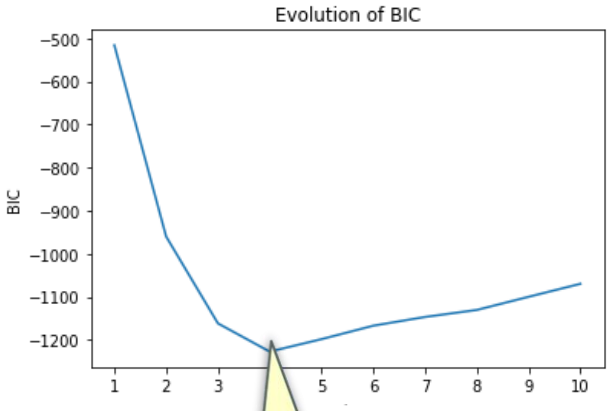


Selectarea valorii k

- Aceeași ca la metoda *k-means*: intuitiv, folosind cunoștințele asupra domeniului, în funcție de un obiectiv
- Criteriul informației bayesiene (BIC – Bayesian Information Criterion)
  - Diferență între complexitatea modelului și probabilitatea clusterelor
  - $BIC = \ln(|X|)\, k' - \hat{L}(C_1, \dots, C_k; X)$
    - k' este numărul de parametri ai modelului (valori medii, covarianțe)
    - $\hat{L}(C_1, \dots, C_k; X) = p(C_1, \dots, C_k|X)$ este funcția de probabilitate
- Cu cât valoarea criteriului este mai mică, cu atât mai bine
  - Valoarea descrește pentru modelele mai puțin complexe
  - Valoarea descrește odată ce probabilitatea este mai bună

Rezultate pentru k=1,…,4

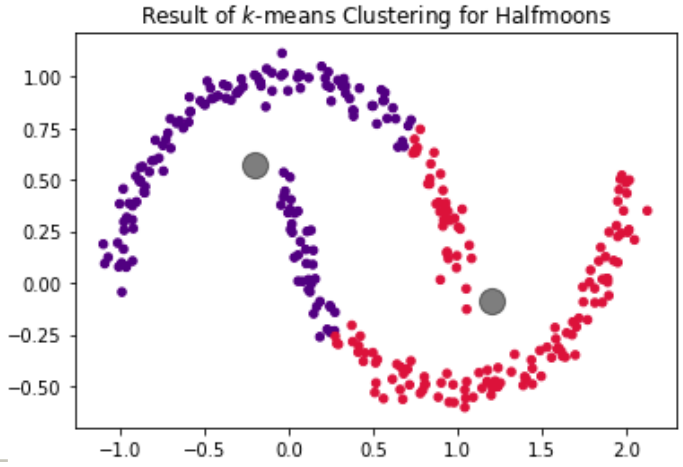




Valoarea minimă = raportul optim dintre probabilitatea modelului și concordanță (*goodness of fit*)

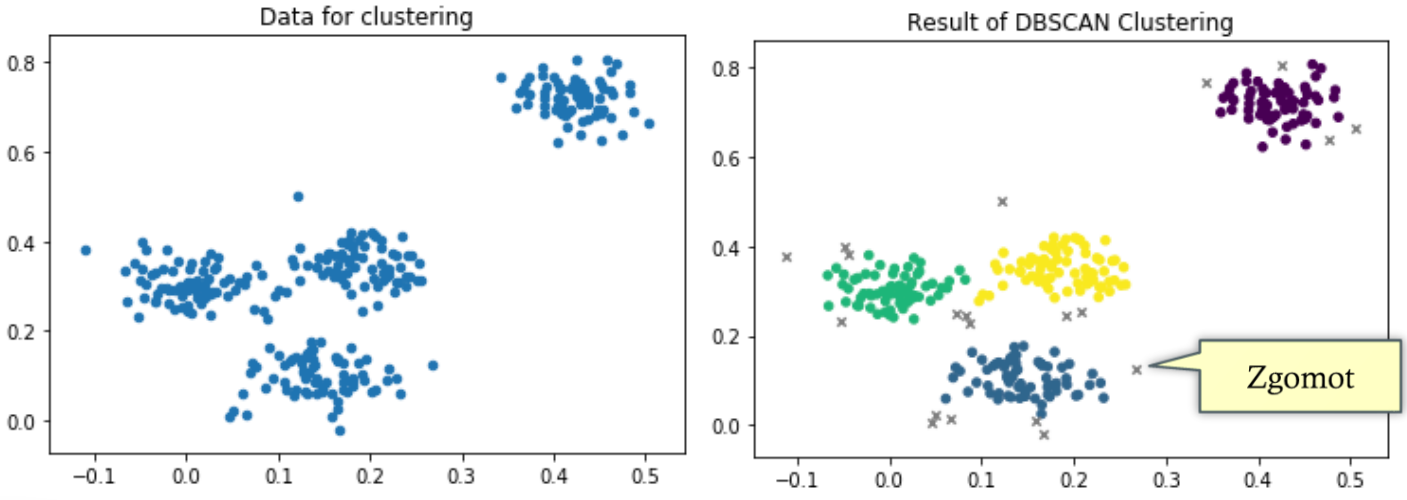
Probleme

- Depinde de clusterele inițiale
  - Rezultatele pot fi instabile
- Alegerea greșită a lui *k* poate conduce la rezultate incorecte
- Poate să nu conveargă
- Funcționează bine doar cu clustere normal repartizate



DBSCAN

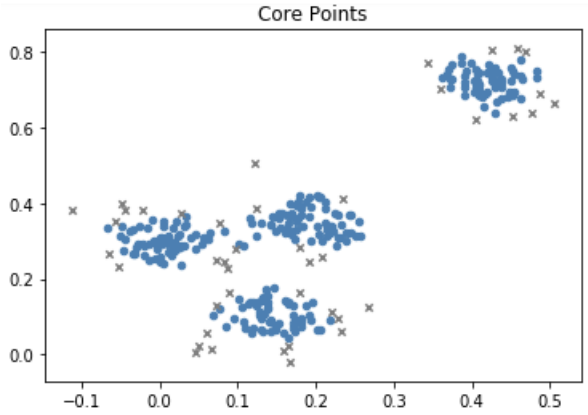
- Clusterele sunt descrise prin intermediul altor obiecte apropiate
  - Clustering bazat pe densitate
- Suprafață de scanare în jurul unui obiect, pertru alte obiecte
  - Dacă sunt găsite obiecte, atunci probabil ele aparțin aceluiași grup
  - Dacă nu sunt găsite obiecte, obiectul este probabil doar un „zgomot”(*noise*)

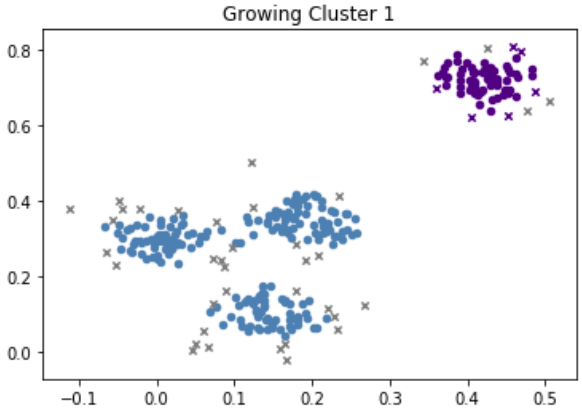


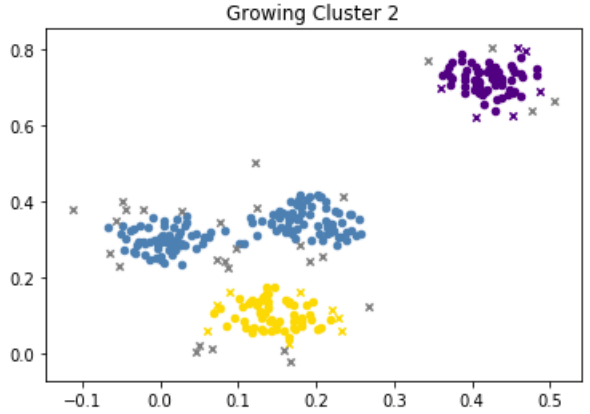
Algoritm

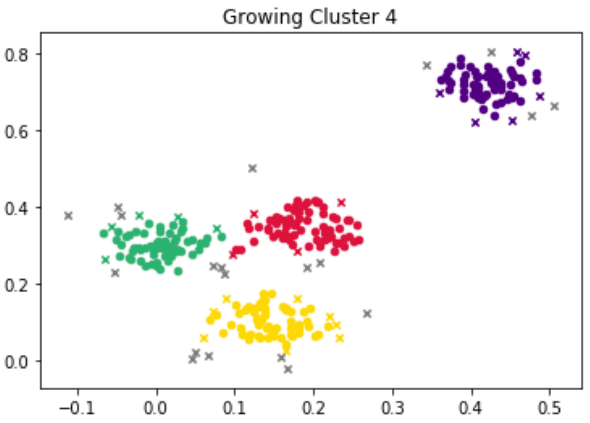
- 2 parametri
  - Dimensiunea vecinătății ε
  - Numărul minimal de puncte pentru a fi considerată densă *minPts*
- Se determină toate obiectele cu vecinătăți dense (puncte nucleu)
  - $x \in X$ astfel încât $|\{x' \in X | d(x, x') \leq \varepsilon\}| \geq minPts$
- Clusterele se măresc prin asignarea în același cluster a tuturor punctelor care partajează o vecinătate
- Toate punctele care nu sunt nici puncte nucleu, nici nu se află în vecinătatea unui punct nucleu constituie zgomot

Vizualizarea algoritmului



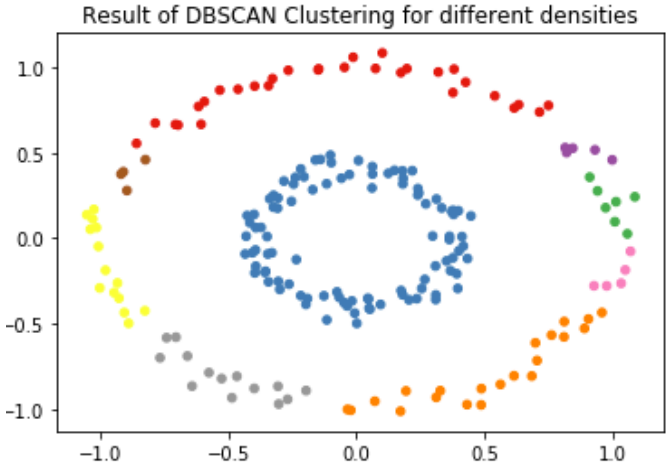






Probleme

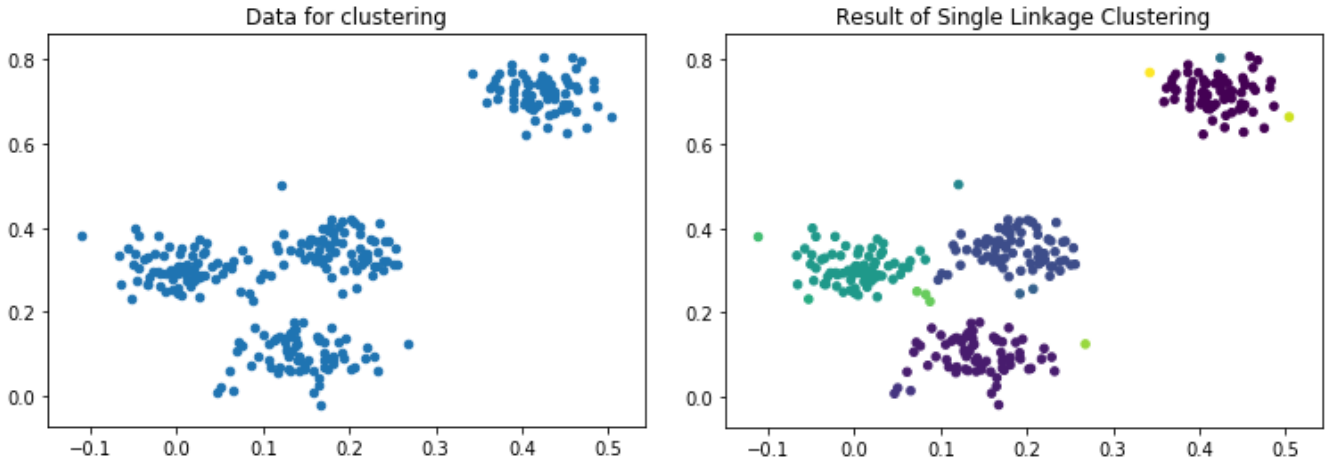
- Toate caracteristicile trebuie să se afle în același domeniu de valori
- Ce se întâmplă dacă clustere diferite au densități diferite?
  - Problema principală a lui DBSCAN



- Această problemă are legătură și cu dimeniunea datelor
  - DBSCAN este foarte sensibil la eșantionare (*sampling*)

Clustering ierarhic

- Clusterele sunt descrise prin ierarhii de similaritate
  - Clustering ierarhic (denumit și clustering bazat pe conectivitate)
- Se găsește cea mai similară pereche de obiecte și se stabilește o legătură
  - Clustering „Nearest neighbor”

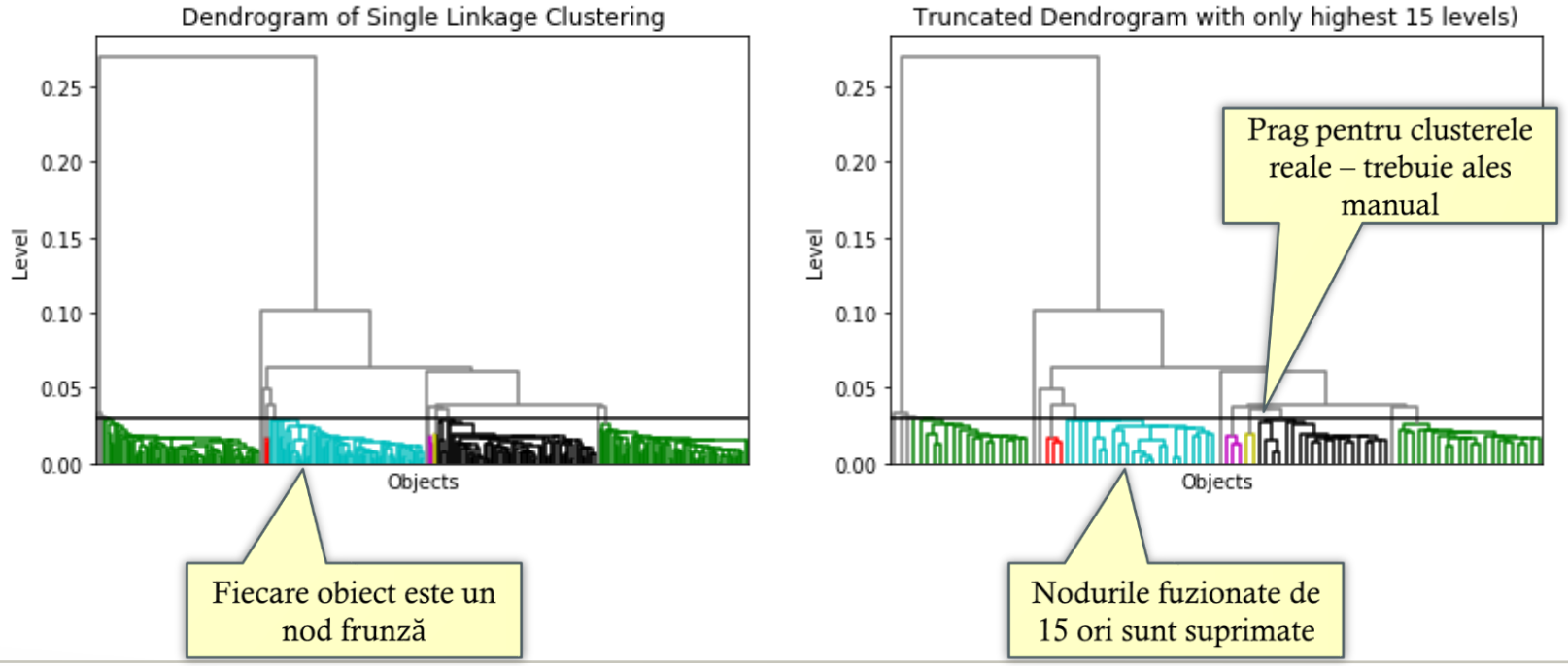


Algoritmul *Single Linkage* (SLINK)

- Fiecare obiect are propriul său cluster la început
- Nivelul (*level*) al acestor clustere de bază este 0
  - $L(C) = 0$ pentru toți $C = \{x\}$, unde $x \in X$
- Se găsesc 2 cele mai apropiate clustere
  - $C, C' = \underset{C,C' \in Clusters}{\operatorname{argmin}} \; d(C, C')$
  - $d(C, C') = \min_{x \in C, x' \in C} d(x, x')$
- C, C' vor fuziona într-un nou cluster $C_{new} = C \cup C'$
- Nivelul (*level*) noului cluster este egal cu distanța dintre clusterele inițiale
  - $L(C_{new}) = d(C, C')$

Vizualizarea SLINK

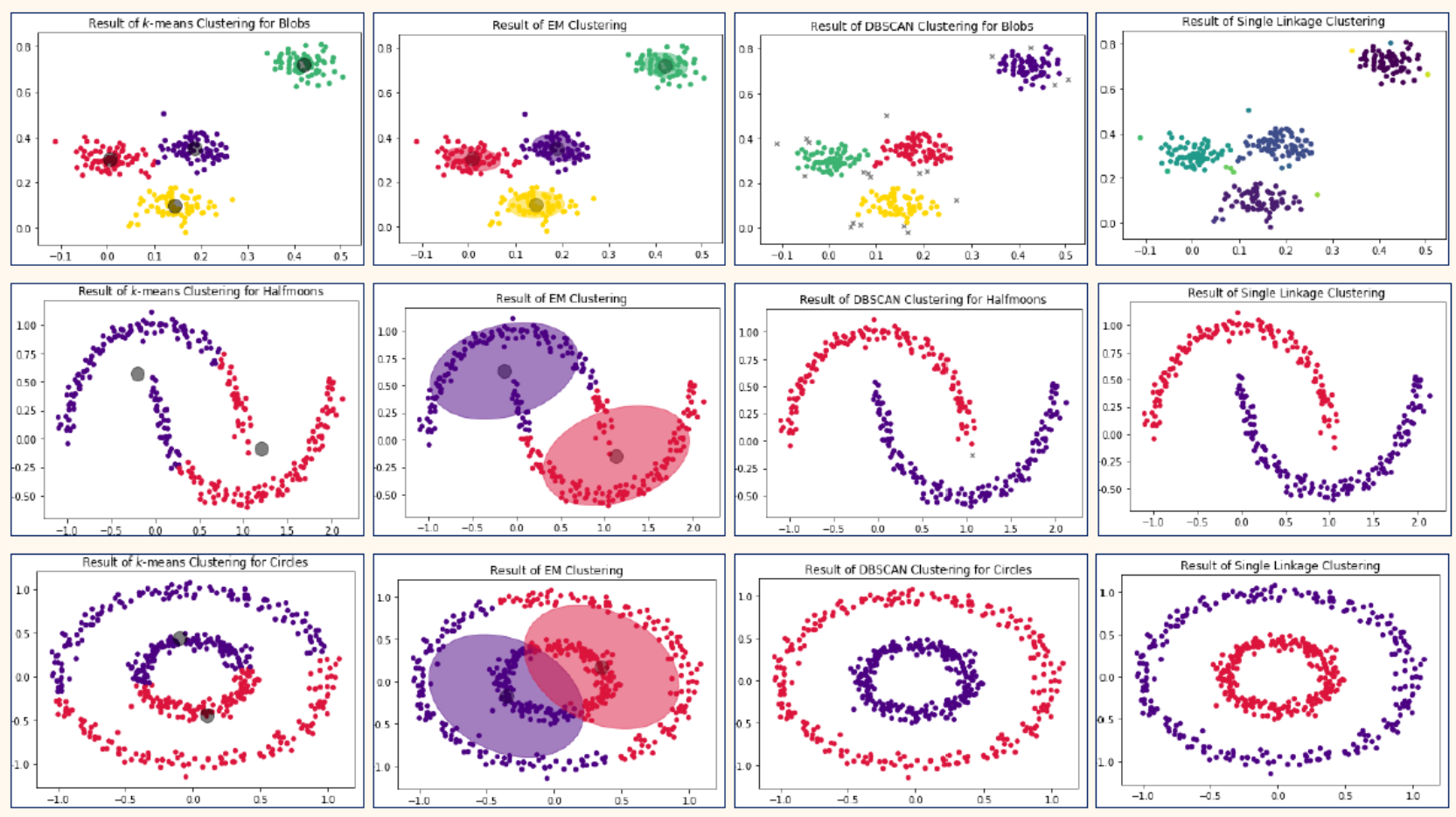
- Dendrogramele permit vizualizarea clustering-ului sub formă de arbore
  - Linia orizontală: Fuziunea a 2 clustere
  - Linia verticală: Creșterea nivelului datorată fuziunii



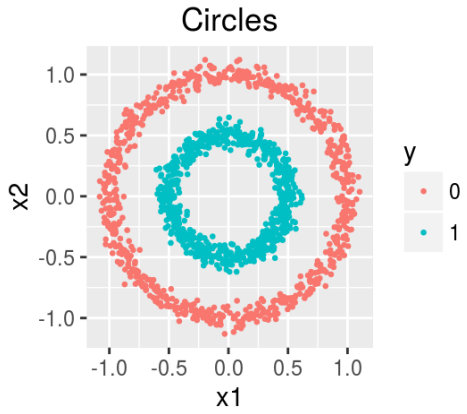
Probleme

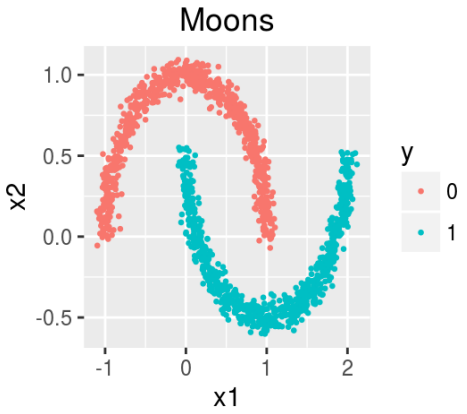
- Clusteringul ierarhic scalează adeseori prost în ceea ce privește consumul de memorie
  - Algoritmul standard necesită matrici pătratice de distanțe între toate obiectele
- Toate caracteristicile trebuie să se afle în același domeniu de valori
- Pot fi problematice densitățile diferite în clustere diferite
  - Este dificil de găsit un unic prag (*cut-off*)
  - Se poate rezolva prin analiza vizuală a dendogramei

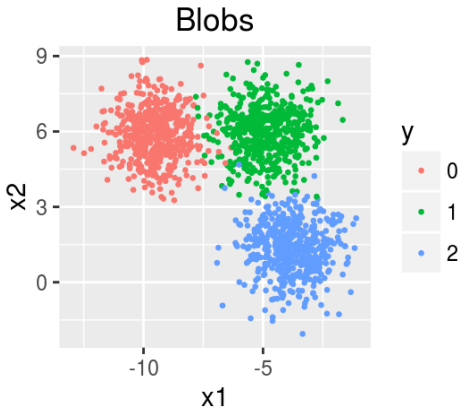
Comparație a clusterelor



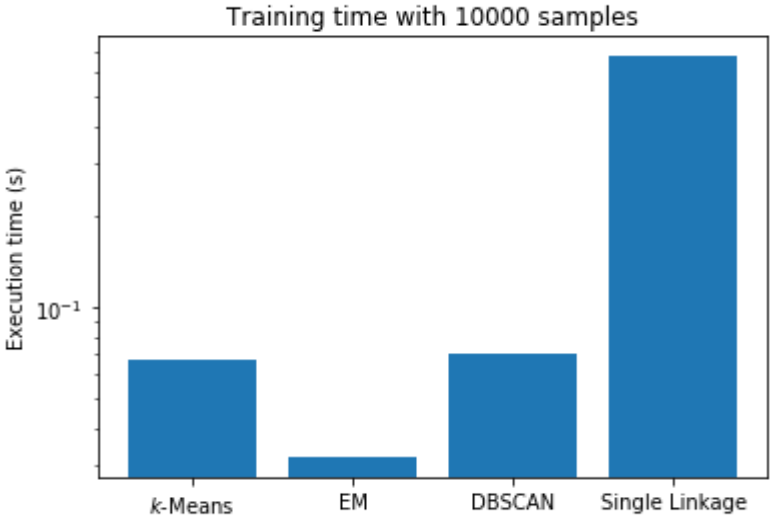
Comparație a clusterelor

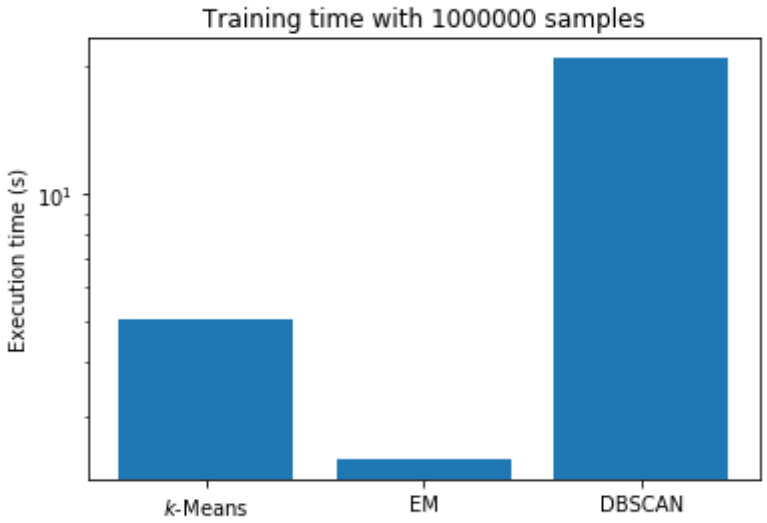






Comparație a timpilor de execuție





- SLINK necesită foarte multă memorie pentru clustere mari

Avantaje și dezavantaje

	Număr de clustere	Valoare explicativă	Reprezentare concisă	Caracteristici categoriale	Caracteristici lipsă	Caracteristici corelate
k-means	-	+	+	-	-	0
EM	0	+	+	-	-	0
DBSCAN	+	-	-	-	-	0
SLINK	0	+	-	-	-	0

Există algoritmi de clustering pentru date categoriale, de exemplu *k-modes*

Concluzii

- Clusteringul se referă la inferența grupărilor de obiecte
- Funcționează bine pentru date numerice dar adeseori nu este adecvat datelor categoriale
  - Scalele sunt foarte importante pentru majoritatea algoritmilor de clustering
- Diferite tipuri de algoritmi de clustering:
  - Algoritmi bazați pe centroizi
  - Algoritmi bazați pe repartiții (distribuții)
  - Algoritmi bazați pe densitate
  - Algoritmi bazați pe conectivitate (ierarhici)
- Evaluarea este de multe ori dificilă și necesită intervenție manuală

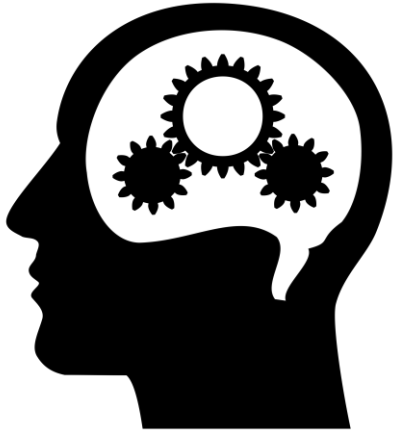
6. Clasificare

Clasificare

- Descriere
- Modele de clasificare
- Compararea modelelor de clasificare
- Concluzii

Exemplu de clasificare

Aceasta este o balenă

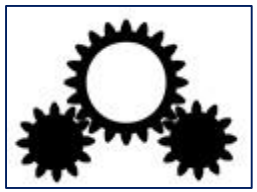


Acesta este un urs

Problema generală

Obiect 1

Clasă de obiecte 1



Concept

Obiect 2

Clasă de obiecte 2

Problema formală

- Spațiu de obiecte
  - $O = \{object_1, object_2, \dots\}$
  - Poate fi infinit
- Reprezentări ale obiectelor într-un spațiu de caracteristici
  - $\mathcal{F} = \{\phi(o)\ ,\ o \in O\}$
- Mulțime de clase
  - $C = \{class_1, \dots, class_n\}$
- O funcție (*target concept*) ce mapează obiectele la clase
  - $h^*: O \rightarrow C$

Cum se obține h^*?

- Clasificare
  - Determinarea unei aproximări a funcției

Ipoteze

- Cum recunoaștem o poză cu o balenă?

Ipoteză: Obiectele care au o aripă dorsală, o formă generală ovală, care sunt negre în partea de sus și albe în cea de jos și care se află pe un fundal albastru sunt balene.

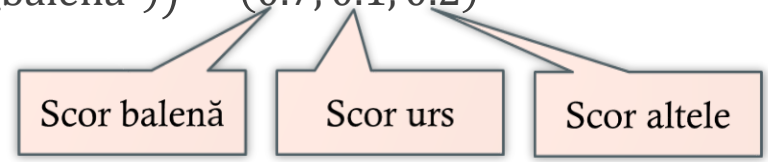
Ipoteze

- O ipoteză mapează caracteristici la clase
  - $h: \mathscr{F} \rightarrow C$
  - $h: \phi(o) \rightarrow C$
- Aproximare a funcției h^*
  - $h^*(o) \approx h(\phi(o))$
- Ipoteza = Clasificator = Model de clasificare

Ce se întâmplă dacă nu suntem siguri în privința claselor?

Clasificarea pe bază de scoruri

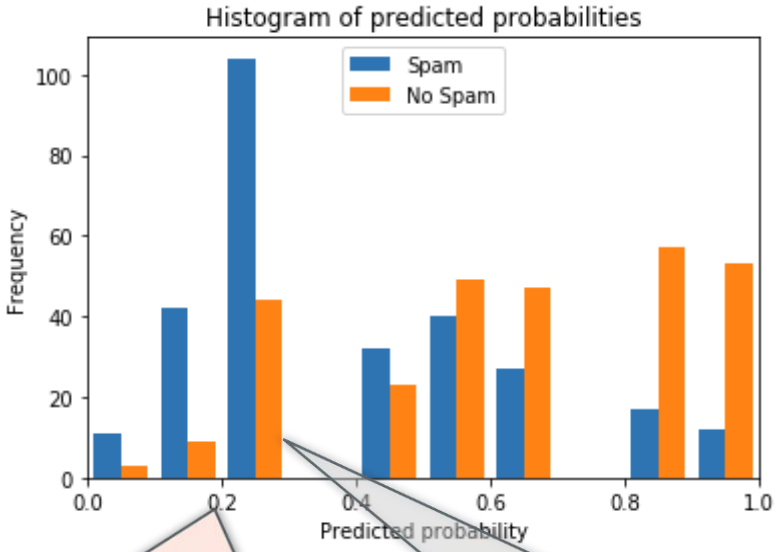
- Un scor numeric pentru fiecare clasă $c \in C$
- De cele mai multe ori, o distribuție de probabilitate
  - $h' : \phi(o) \rightarrow [0,1]^{|C|}$
  - $\|h'(\phi(o))\|_1 = 1$
- Exemplu:
  - 3 clase: balenă, urs, altele
  - $h'\big(\phi(\text{"poza\_balena"})\big) = (0.7, 0.1, 0.2)$



- Abordarea standard:
  - Clasificarea este reprezentată de clasa având cel mai mare scor

Praguri pentru scoruri

- Este posibilă definirea de diferite praguri



Pragul de 0.2 ar putea rata clasificarea unor mesaje de tip „spam” dar le-ar putea identifica mai bine pe cele „no spam”

Multe mesaje de tip „no spam” detectate incorect dacă este folosit cel mai mare scor

Calitatea ipotezelor

- Scop: Aproximarea funcției
  - $h^*(o) \approx h(\phi(o))$

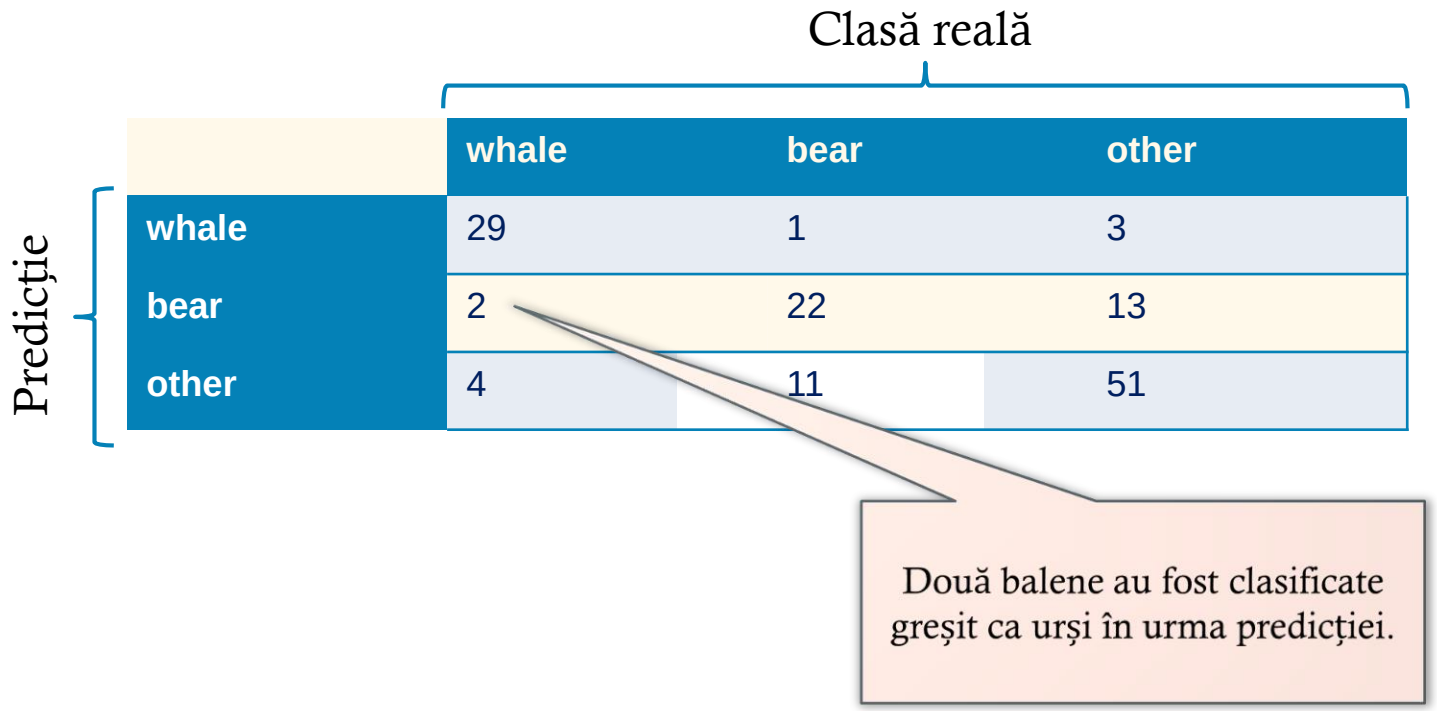
Cum se evaluează $h^*(o) \approx h(\phi(o))$?

➢ Se folosesc date de test
  - Structura este aceeași cu a datelor de training
  - Se aplică ipoteze

$\phi(o)$					$h^*(o)$	$h(\phi(o))$
hasFin	shape	colorTop	colorBottom	background	class	prediction
true	oval	black	black	blue	**whale**	**whale**
false	rectangle	brown	brown	green	**bear**	**whale**
…	…	…	…	…	…	

Matricea de confuzie

- Tabel al valorilor reale versus predicție

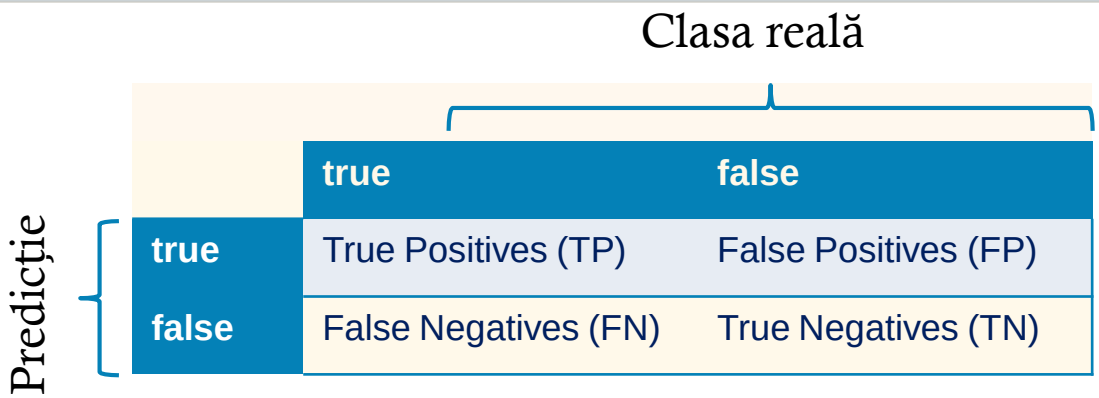


Clasificare binară

- Multe probleme sunt binare
  - Îmi primesc banii înapoi?
  - Un anumit card de credit este fraudulos?
  - Etc.
- Pot fi formulate în termeni de apartenență (sau nu) la o clasă

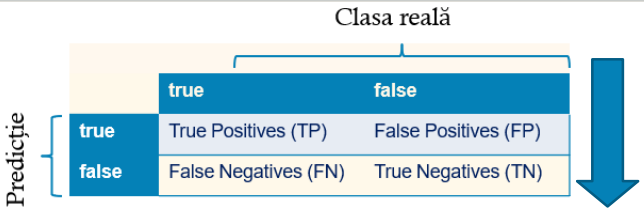
➢ Etichetele *true* și *false*

Matricea de confuzie binară



- Valorile False Positive (FP) se mai numesc eroare de tipul I
- Valorile False Negative (FN) se mai numesc eroare de tipul II

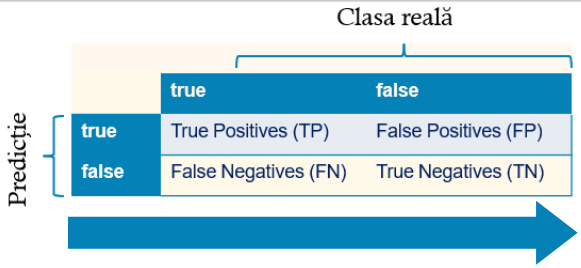
Metrici de performanță binară (1)



- Rate în funcție de clasa reală:
  - Rata valorilor True Positive (*recall*, *sensitivity*)
    - Procentul valorilor reale True care sunt prezise corect
    - $TPR = \frac{TP}{TP+FN}$
  - Rata valorilor True Negative (*specificity*)
    - Procentul valorilor reale False care sunt prezise corect
    - $TNR = \frac{TN}{TN+FP}$
  - Rata valorilor False Negative
    - Procentul valorilor reale True care sunt prezise greșit
    - $FNR = \frac{FN}{FN+TP}$
  - Rata valorilor False Positive
    - Procentul valorilor reale False care sunt prezise greșit
    - $FPR = \frac{FP}{FP+TN}$

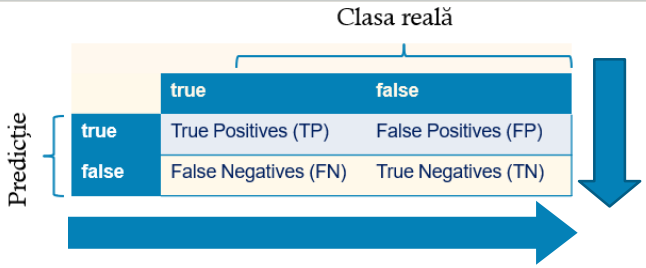
Metrici de performanță binară (2)

- Rate în funcție de clasa prezisă:
  - Valoare pozitivă predictivă (precizie)
    - Procentul predicțiilor True care sunt prezise corect
    - $PPV = \frac{TP}{TP+FP}$
  - Valoare predictivă negativă
    - Procentul predicțiilor False care sunt prezise corect
    - $NPV = \frac{TN}{TN+FN}$
  - Rata de descoperire falsă
    - Procentul predicțiilor True care sunt prezise greșit
    - $FDR = \frac{FP}{FP+TP}$
  - Rata omisiunilor false
    - Procentul predicțiilor False care sunt prezise greșit
    - $FOR = \frac{FN}{FN+TN}$



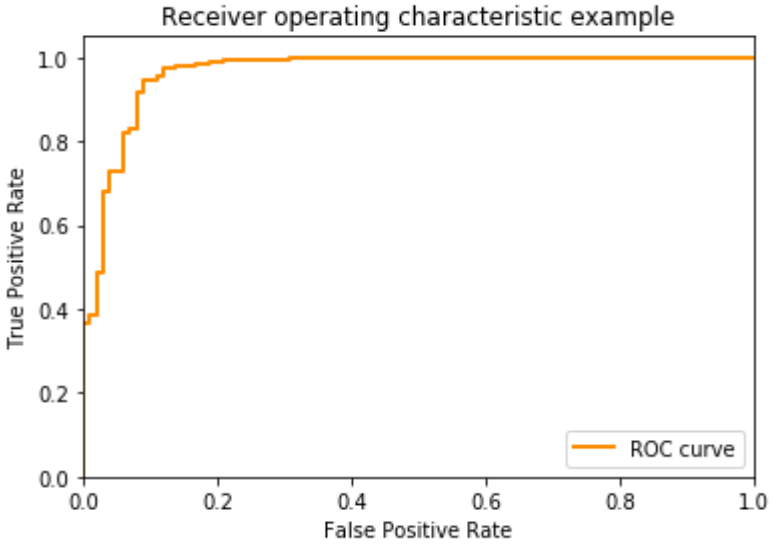
Metrici de performanță binară (3)

- Metrici care iau totul în considerare
- Acuratețea
  - Procentul datelor prezise corect
  - $accuracy = \frac{TP+FN}{TP+TN+FP+FN}$
- Măsura F1
  - Medie armonică a preciziei și *recall*-ului
  - $F_1 = 2\frac{precision \times recall}{precision+recall}$
- Coeficientul de corelare Matthews (MMC)
  - Corelare χ^2 dintre predicție și valorile reale
  - $MMC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$



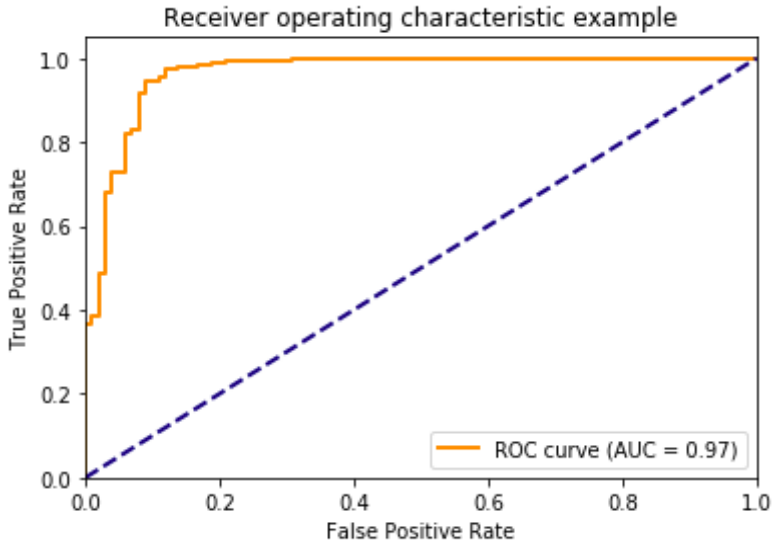
Receiver Operator Characteristics (ROC)

- Grafic al ratei valorilor True pozitive (TPR) versus rata valorilor False pozitive (FPR)
- Este posibil să se obţină valori diferite TPR/FPR din cauza pragurilor pentru scoruri



Area Under Curve (AUC)

- Cu cât aria de sub curbă este mai mare, cu atât performanța este mai bună



Micro și *macro Averaging*

- Metrici neaplicabile direct pentru un număr de clase mai mare decât 2
  - Acuratețea este o excepție
- *Micro averaging*
  - Extinde formulele anterioare luând în calcul exemple pozitive și negative individuale pentru fiecare clasă
- *Macro averaging*
  - Consideră o clasă ca fiind True și le combină pe toate celelalte ca False
  - Calculează metrici pentru toate aceste combinații
  - Determină media
- Exemplu pentru rata valorilor True pozitive:
  - $TPR_{micro} = \frac{\sum_{c \in C} TP_c}{\sum_{c \in C} TP_c + \sum_{c \in C} FN_c}$
  - $TPR_{macro} = \frac{\sum_{c \in C} \frac{TP_c}{TP_c + TN_c}}{|C|}$

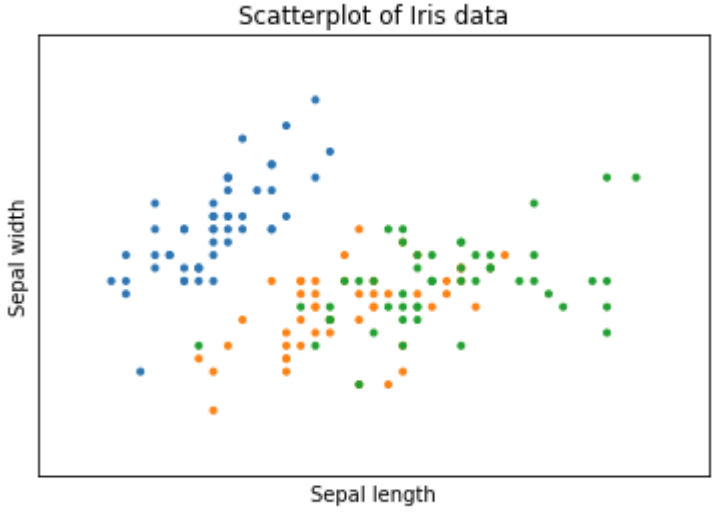
Modele de clasificare

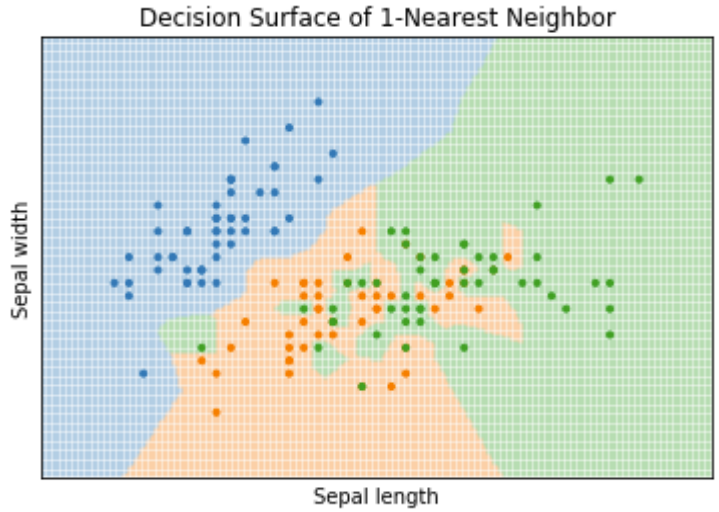
- Sunt prezentați următorii clasificatori
  - *k-Nearest Neighbor*
  - Arbori de decizie
  - *Random Forests*
  - Regresie logistică
  - *Naive* Bayes
  - SVM (*Support Vector Machines*)
  - Rețele neuronale

k-Nearest Neighbor

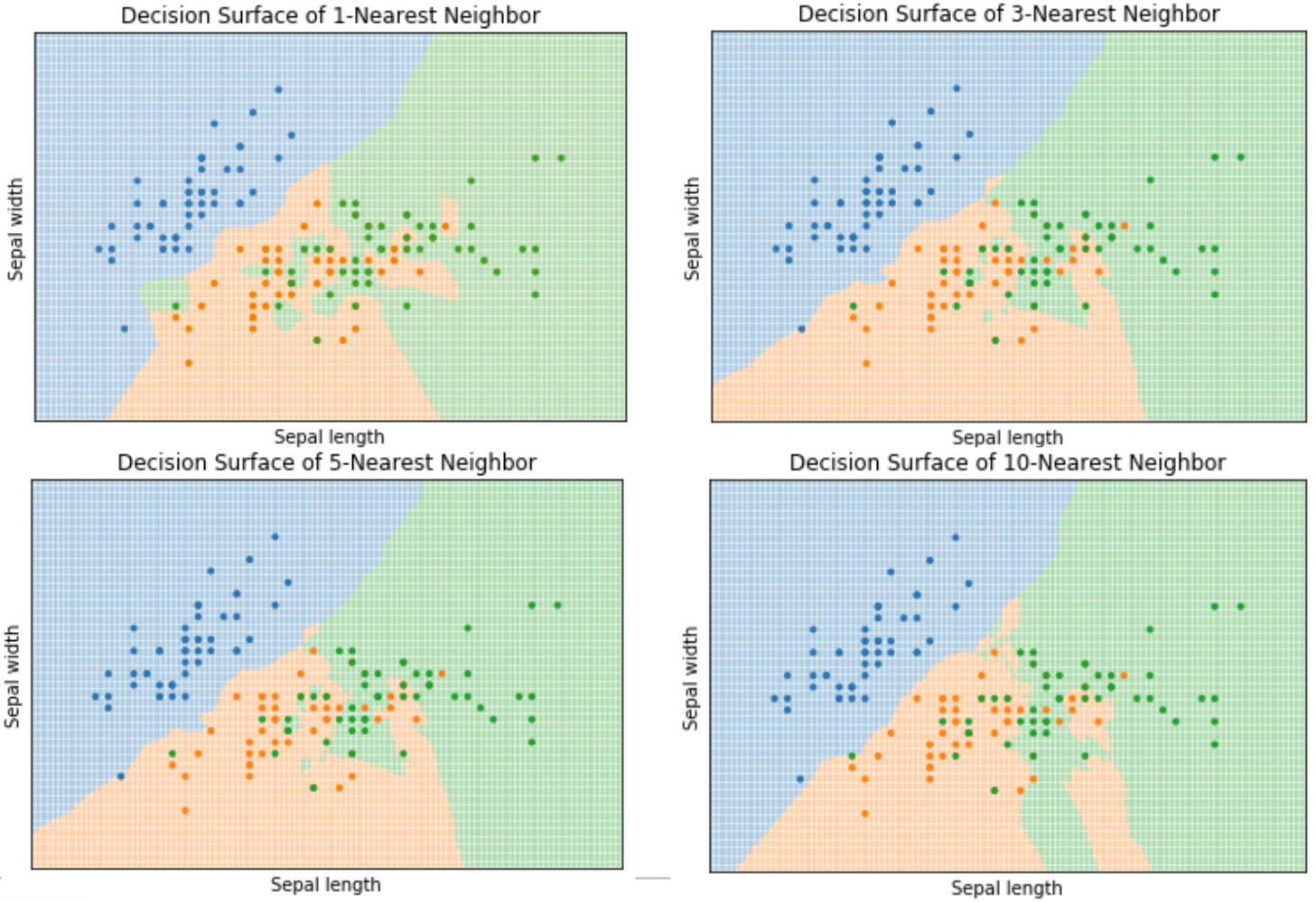
- Ideea de bază
  - Instanțe cu valori similare ale caracteristicilor ar trebui să aparțină aceleiași clase
  - Clasa poate fi determinată analizând instanțele care sunt similare

➢ Se asignează fiecărei instanțe modul celor *k* cele mai apropiate instanțe



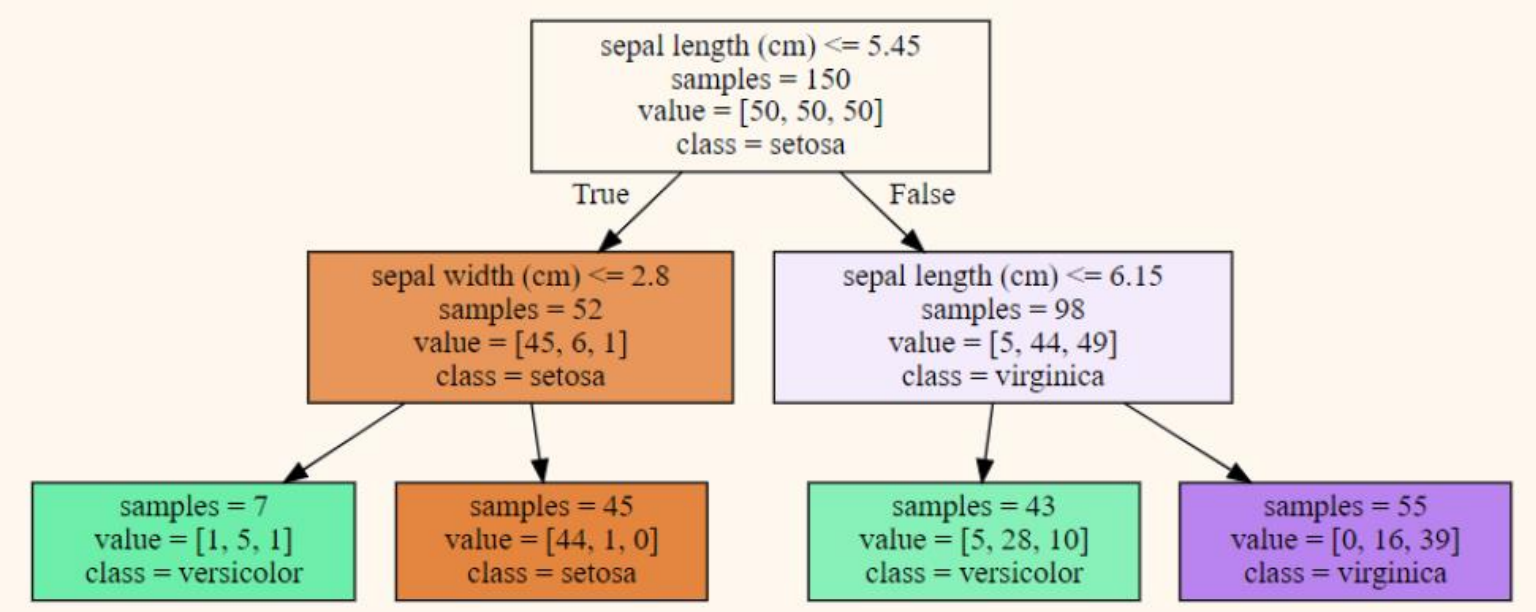


Impactul valorii *k*



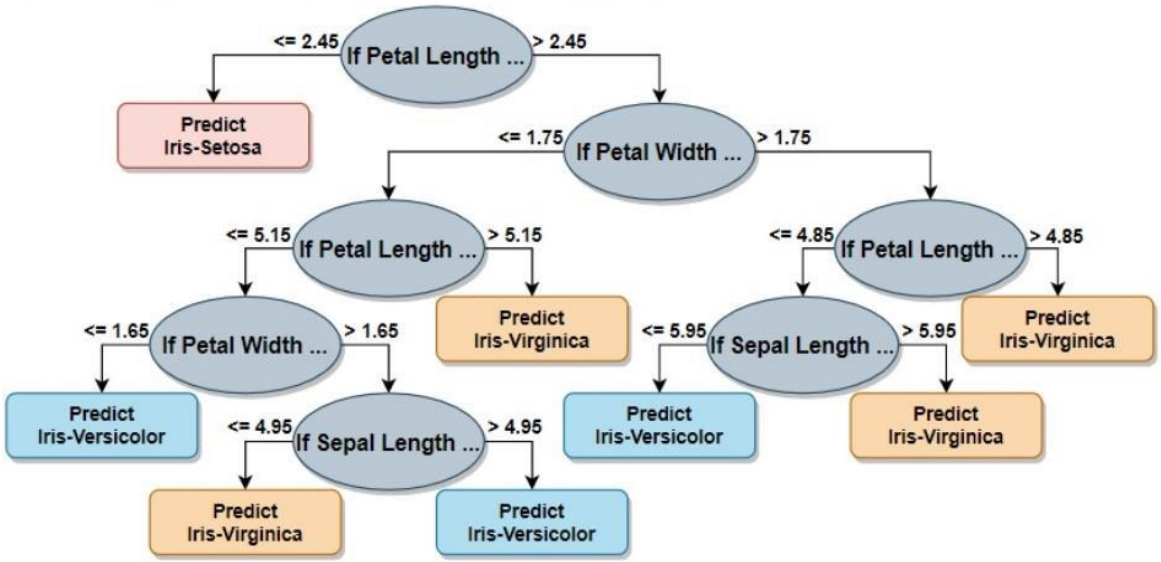
Arbori de decizie

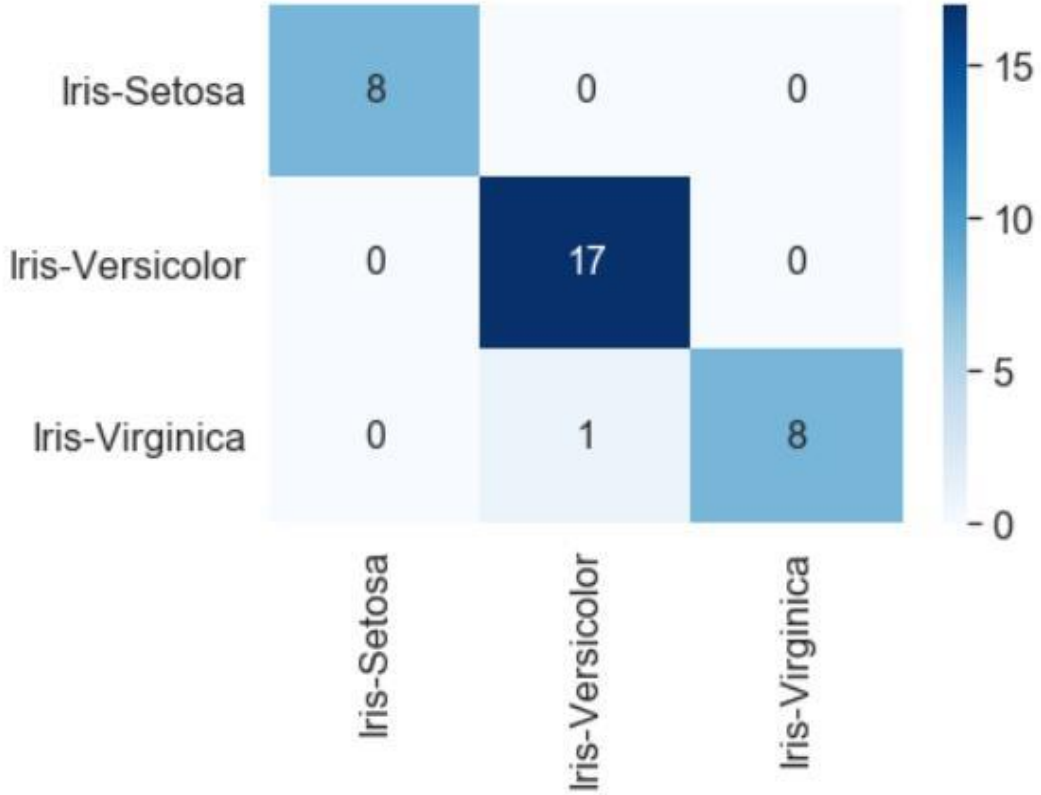
- Ideea de bază
  - Luarea deciziilor pe baza regulilor logice asupra caracteristicilor
  - Regulile sunt organizate sub formă de arbore



```
decision_tree_model = DecisionTreeClassifier(featuresCol = 'Features',
                                             labelCol = 'Class')   \
                        .fit(train_data)
print(decision_tree_model.toDebugString)
```

```
DecisionTreeClassificationModel (uid=DecisionTreeClassifier_7fe4429a3ea8) of depth 5 with 15 nodes
  If (feature 2 <= 2.45)
   Predict: 0.0
  Else (feature 2 > 2.45)
   If (feature 3 <= 1.75)
    If (feature 2 <= 5.15)
     If (feature 3 <= 1.65)
      Predict: 1.0
     Else (feature 3 > 1.65)
      If (feature 0 <= 4.95)
       Predict: 2.0
      Else (feature 0 > 4.95)
       Predict: 1.0
    Else (feature 2 > 5.15)
     Predict: 2.0
   Else (feature 3 > 1.75)
    If (feature 2 <= 4.85)
     If (feature 0 <= 5.95)
      Predict: 1.0
     Else (feature 0 > 5.95)
      Predict: 2.0
    Else (feature 2 > 4.85)
     Predict: 2.0
```




Iris-Setosa
Iris-Versicolor
Iris-Virginica
8
0
0
0
17
0
0
1
8
15
10
5
0
Iris-Setosa
Iris-Versicolor
Iris-Virginica

Algoritmul de bază

- Algoritm recursiv
  - Se oprește dacă
    - Datele sunt „pure”, adică în mare parte din clasă
    - Cantitatea de date este prea mică, adică sunt prea puține instanțe în partiție
  - Altfel
    - Se determină caracteristica cea mai informativă X
    - Se partiționează datele de antrenare folosind X
    - Se creează recursiv un subarbore pentru fiecare partiție
- Detaliile pot fi diferite pentru fiecare algoritm specific
  - De exemplu CART, ID3, C4.5
- Conceptul general se păstrează

Caracteristica cea mai informativă

- *Most Informative Feature*
- Abordare bazată pe teoria informației
- Entropia etichetei clasei
  - $H(C) = -\sum_{c \in C} p(c) \log p(c)$

Poate fi folosită ca măsură a „purității”.

- Entropia condiţională a etichetei clasei pe baza caracteristicii X
  - $H(C|X) = -\sum_{x \in X} p(x) \sum_{c \in C} p(c|x) \log p(c|x)$
- Informația mutuală
  - $I(C, X) = H(C) - H(C|X)$

Fiecare dimensiune este interpretată ca variabilă aleatoare.

➢ Caracteristica având cea mai mare informație mutuală este cea mai informativă.

Suprafața de decizie a arborilor de decizie

- Toate deciziie sunt aliniate pe axe