Arbori de decizie și regresie

Ansambluri

Random Forests

Date și model

- Principiul este comun
  - Clasificare
  - Regresie
- Formal: avem datele de antrenament sub formă de **vectorii X_i** cu etichetele Y_i. Etichele sunt**:**
  - Categoriale (discrete) pentru clasificare
  - Continue pentru regresie

Inducție

- Principiul **inducției**:
  - Extragem reguli din exemple
  - Presupunem ca regulile sunt valabile și când avem date foarte multe
- Paradigma inducției și deducției:
  - În pasul inductiv formăm regulile
  - În pasul deductiv, folosim regulile pentru a prezice etichete pentru datele noi

Arbori de clasificare și regresie

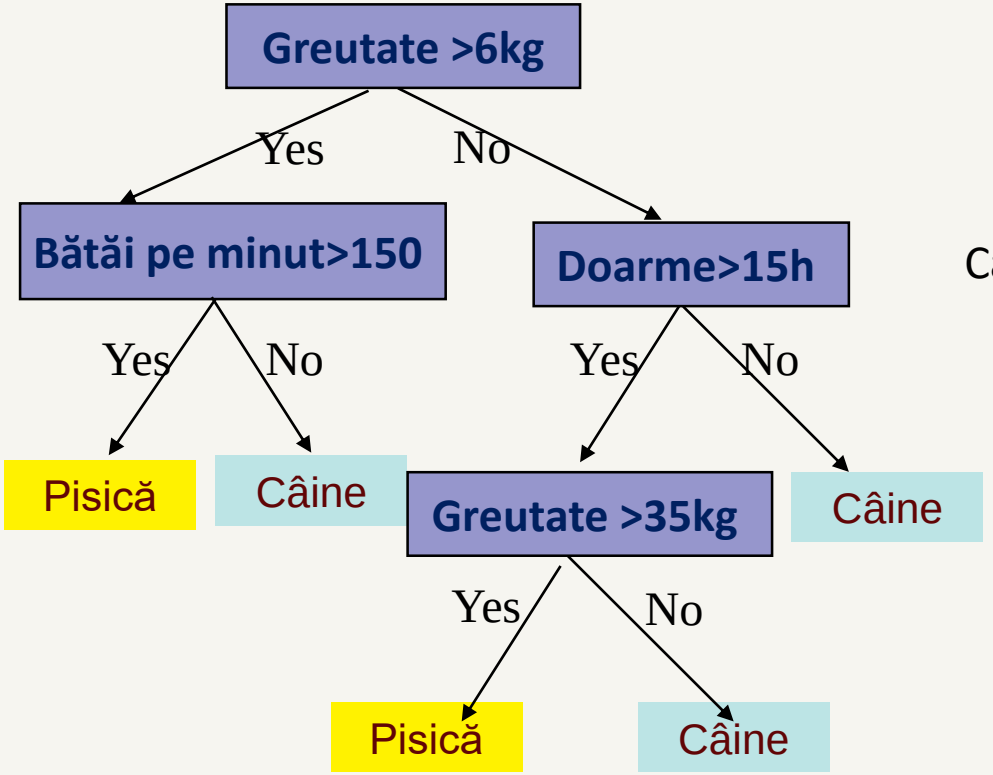
- Un arbore este un model predictiv care:
  - Se construiește pe baza unui set de decizii binare
  - Calculează o valoare de ieșire

- Diferența între regresie și clasificare (la construcție)
- este dată de funcția obiectiv
-

Ce este un arbore de decizie?

- Folosește abordare ***inductivă***
  - Folosește date particulare pentru a construi reguli mult mai generale
- Un model predictiv bazat pe o serie de teste boolene
  - Succesiunea de teste este mai puternică decât mulți clasificatori complecși
- Cum arată un arbore de decizie

Animalul acesta este ... Pisică sau Câine



Câinii sunt mai masivi, dar

Există pisici obeze și există chihuahua

Câinii f. mari dorm mult

Animal = (greutate, bătăi pe minut, cât doarme, indice de frumusețe)

indice de frumusețe – nu e util

Ce animal e cel descris de (45,80, 10 9)

Dar

(8,180,18,7)

Învățare Inductivă

- În acest arbore de decizie, am făcut o serie de decizii binare și am construit o ramură
  - Un animal: ce gretate are?
  - Cât doarme?
  - Ce ritm cardiac are?

- Răspunzând la aceste intrebări cu DA sau NU, facem diferența între câini și pisici

Datele într-un tabel

Setul de antrenament

Exemplu	**Atributele**				**Eticheta**
	Greutate	Ritm cardiac	Cât doarme	Frumusete	
Lăbuș	25	100	8	5	Câine - labrador
Puffy	3.5	180	16	9	Pisică - europeana
Max	65	45	13	7	Câine ciobanesc
Rex	6	130	16	8	Câine canis
Dingo	2	200	15	7	Pisică - slabanog
Brutus	1.5	140	7	1	Câine - pechinez
Asci	15	160	19	8	Pisică - maine coon gras
Mutzi	12	130	20	2	Pisică - obez
Caramel	5	120	16	9	Pisică - birmaneza
Blacky	4	220	16	10	Pisică - norvegiana
Neige	20	80	18	10	Câine - Husky
Garfield	8	180	19	4	Pisică - roscata
Toto	30	85	12	6	Câine - corcitura

Alegerea atributelor

- Tabelul anterior arătă 4 atribute: greutate, ritm cardiac, durată somn și frumusețe
- Dar decizia este luată pe baza doar a trei
  - Frumusețea nu e relevantă
- De ce? E bine?

Cum se creează un Arbore de decizie

- Datele sunt descrise de o listă de atribute
  - Atributele pot fi discrete sau continue
- Se consideră pe rând fiecare atribut și pentru momentul curent se alege cel care produce cea mai bună împărțire
- Se fixează un prag și se obțin două subprobleme care se rezolvă recursiv similar

Construcția unui arbore

- Antrenare
- Ce variabile se folosesc în comparația actuală și unde?
- Când ne oprim? Continuăm?
- Nodul terminal primește o etichetă.

Algorim pentru arbore de decizie

- Ideea de bază este :
  - Se alege ***cel mai bun*** atribut pentru comparație și se împart exemplele după decizia luată, pe baza acelui atribut
  - Se repetă procesul, recursiv, pentru fiecare sub-arbore
  - Ne oprim când :
    - Toate instanțele rămase într-o subproblemă au aceeași etichetă
    - Nu mai sunt atribute de încercat
    - Nu mai sunt date

Clasificare

- Măsura de optimizat:
- Index GINI (index de impuritate)

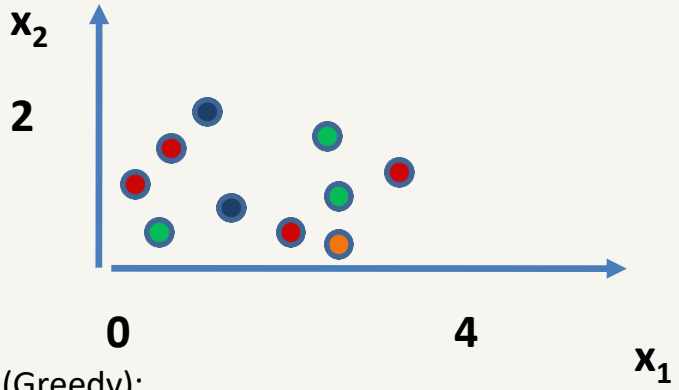
$$GINI(X) = 1 - \sum_{i=1}^{N} (p_i)^2$$

- P_i frecvență relativă a clasei i în X – (sub) setul de date din split-ul respectiv
- Valorile GINI mai mici sunt mai bune. Gini == 0 – clasă pură
- La origine măsoară dezechilibrul social

Arbore de clasificare (decizie)

Datele de antrenament

Obj	x_1	x_2	y
X_1	0.14	1.6	3
X_2	3.7	1.4	1
X_3	2.4	0.6	2
X_N	0.15	0.87	3

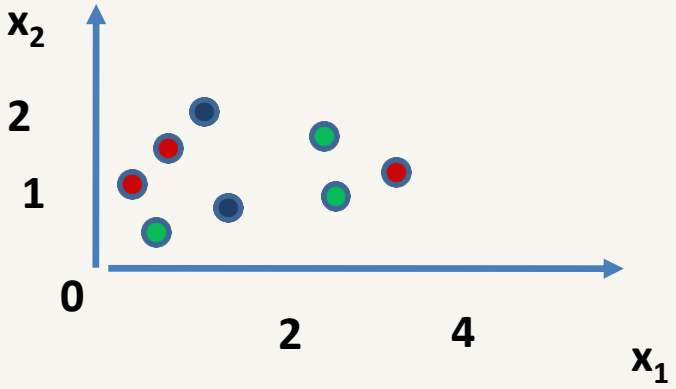


SPLIT (Greedy):
MinGINI = RealMAX
For each dimension d = x_1…x_2
For val = min(d_1 … d_{N-1}): max(d_1 … d_{N-1}
Split between val_{d_i} and val_{d_i+1}
Subset value = the **majority** of values in subset
Compute GINI. If less than MinGINI, store
end
End
Use the dimension and val that lead to MinGINI

Arbore de clasificare (decizie)

Datele de antrenament

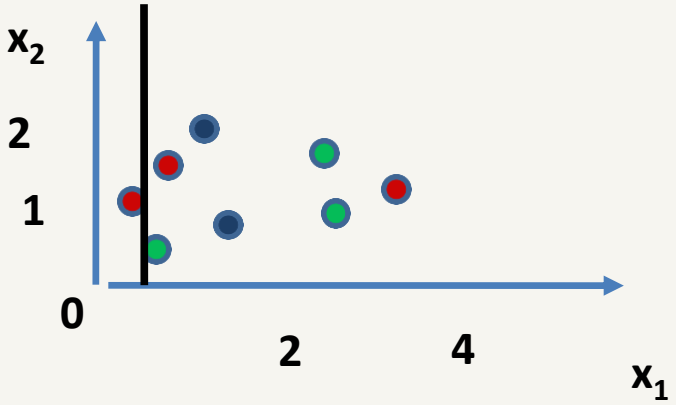
Obj	x_1	x_2	y
X_1	0.14	1.3	3
X_2	3.7	1.4	3
X_3	1.7	0.7	2
X_4	0.5	1.6	3
X_5	1.5	2.2	2
X_6	0.27	0.3	1
X_7	2.4	1.8	1
X_8	2.7	0.87	1



Arbore de clasificare (decizie)

Datele de antrenament

Obj	x_1	x_2	y
X_1	0.14	1.3	3
X_2	3.7	1.4	3
X_3	1.7	0.7	2
X_4	0.5	1.6	3
X_5	1.5	2.2	2
X_6	0.27	0.3	1
X_7	2.4	1.8	1
X_8	2.7	0.87	1



Split $x_1 < 0.2$

Clasa stânga – roșie =3

Clasa dreapta – verde (pluralitate) = 1

$$GINI_{stânga} = 1 - \left(\left(\frac{1}{1}\right)^2 + \left(\frac{0}{1}\right)^2 + \left(\frac{0}{1}\right)^2\right) = 1 - 1 = 0$$

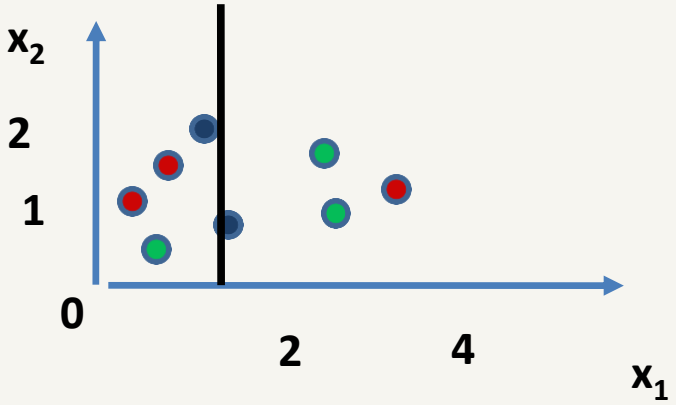
$$GINI_{dreapta} = 1 - \left(\left(\frac{3}{7}\right)^2 + \left(\frac{2}{7}\right)^2 + \left(\frac{2}{7}\right)^2\right) = 0.65$$

$$GINI_{total} = \frac{1}{8} GINI_{stanga} + \frac{7}{8} GINI_{dreapta} = 0.57$$

Arbore de clasificare (decizie)

Datele de antrenament

Obj	x_1	x_2	y
X_1	0.14	1.3	3
X_2	3.7	1.4	3
X_3	1.7	0.7	2
X_4	0.5	1.6	3
X_5	1.5	2.2	2
X_6	0.27	0.3	1
X_7	2.4	1.8	1
X_8	2.7	0.87	1



Split $x_1 < 1.6$

Clasa stânga – roșie =3

Clasa dreapta – verde (pluralitate) = 1

$$GINI_{stânga} = 1 - \left(\left(\frac{1}{4}\right)^2 + \left(\frac{1}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right) = 0.625$$

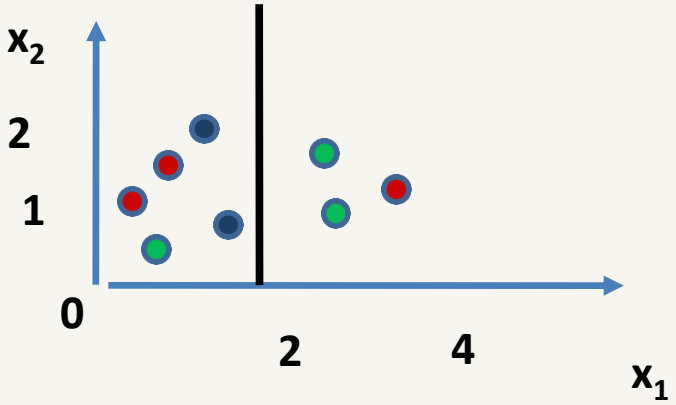
$$GINI_{dreapta} = 1 - \left(\left(\frac{2}{4}\right)^2 + \left(\frac{1}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right) = 0.625$$

$$GINI_{total} = \frac{4}{8} GINI_{stanga} + \frac{4}{8} GINI_{dreapta} = 0.625$$

Arbore de clasificare (decizie)

Datele de antrenament

Obj	x_1	x_2	y
X_1	0.14	1.3	3
X_2	3.7	1.4	3
X_3	1.7	0.7	2
X_4	0.5	1.6	3
X_5	1.5	2.2	2
X_6	0.27	0.3	1
X_7	2.4	1.8	1
X_8	2.7	0.87	1



Split $x_1 < 1.9$

Clasa stânga – abastră =2

Clasa dreapta – verde (pluralitate) = 1

$$GINI_{stânga} = 1 - \left(\left(\frac{1}{5}\right)^2 + \left(\frac{2}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right) = 0.64$$

$$GINI_{dreapta} = 1 - \left(\left(\frac{1}{3}\right)^2 + \left(\frac{0}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right) = 0.44$$

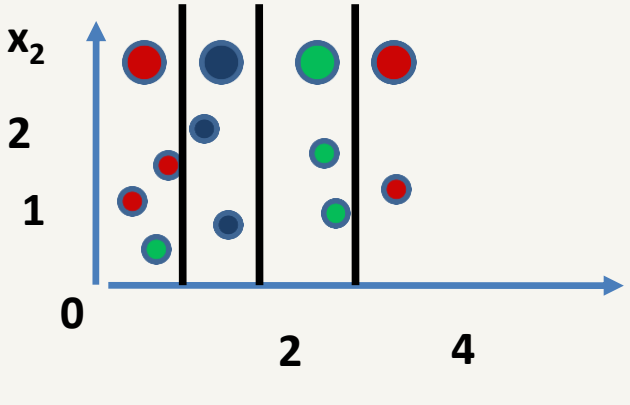
$$GINI_{total} = \frac{5}{8} GINI_{stanga} + \frac{3}{8} GINI_{dreapta} = 0.566$$

Cea mai bună

Arbore de clasificare (decizie)

Datele de antrenament

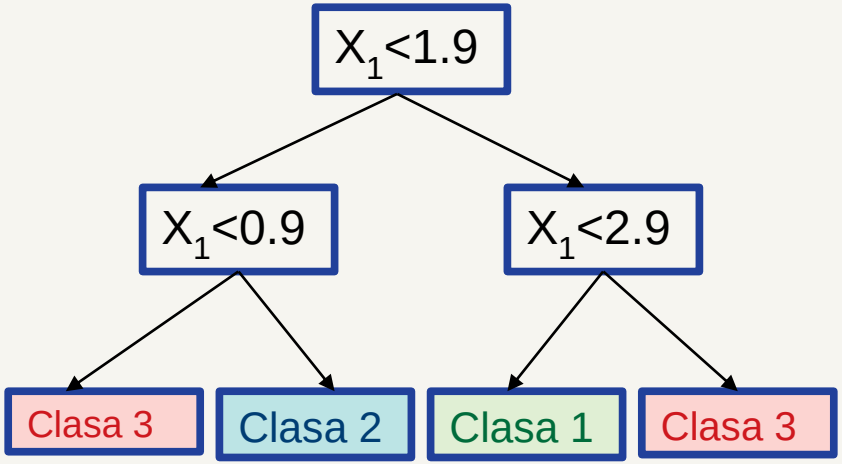
Obj	x_1	x_2	y
X_1	0.14	1.3	3
X_2	3.7	1.4	3
X_3	1.7	0.7	2
X_4	0.5	1.6	3
X_5	1.5	2.2	2
X_6	0.27	0.3	1
X_7	2.4	1.8	1
X_8	2.7	0.87	1

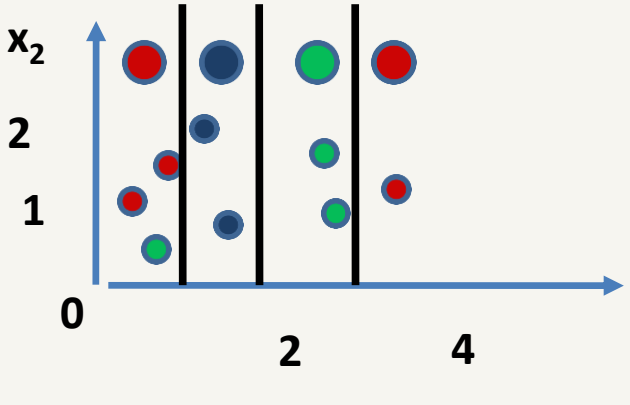


Pentru subarborele stâng split $x_1 < 0.9$

Pentru subarborele drept split $x_1 < 2.9$

Arbore de clasificare (decizie)





Pentru subarborele stâng split $x_1 < 0.9$

Pentru subarborele drept split $x_1 < 2.9$

S-a întamplat ca toate deciziile să fie bazate pe x_1!!!

De obicei aproape toate axele sunt utilizate

Arbore de regresie

- Funcția cost este eroarea pătratică medie:

$$MSE = \sum_{i=1}^{N} \left(Y_i - [Y_i] \right)^2$$

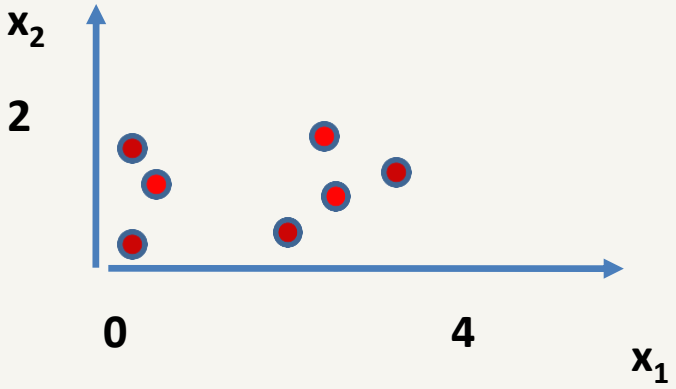
Y_i *eticheta* (adnotarea)

$[Y_i]$ valoarea prezisă de arbore

Arbore de regresie - exemplu

Training data

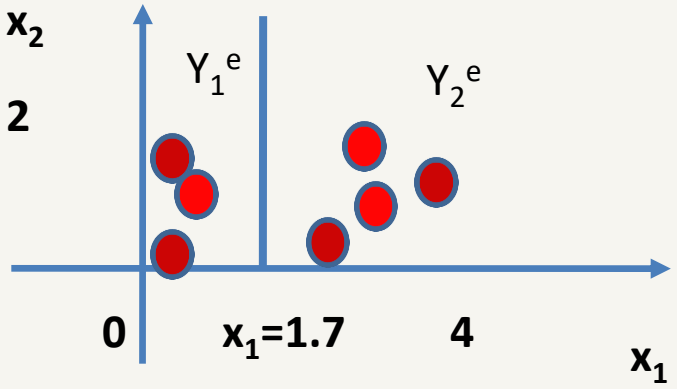
Obj	x_1	x_2	y
X_1	0.14	1.6	0.23
X_2	3.7	1.4	1.90
X_3	2.4	0.6	3.56
X_N	0.15	0.87	1.5

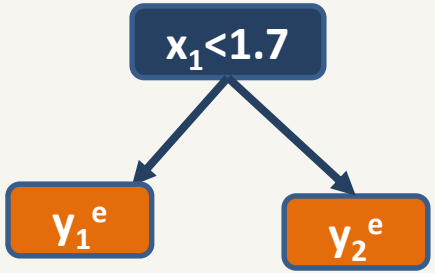


SPLIT (Greedy):
MinMSE = RealMAX
For each dimension d = $x_1 \ldots x_2$
For val = min(d_1 … d_{N-1}): max(d_1 … d_{N-1}
Split between val_{d_i} and val_{d_i+1}
Predicted value = mean of values in split
Compute MSE. If less than MinMSE, store
end
End
Use the dimension and val that lead to MinMSE

Arbore de regresie - exemplu

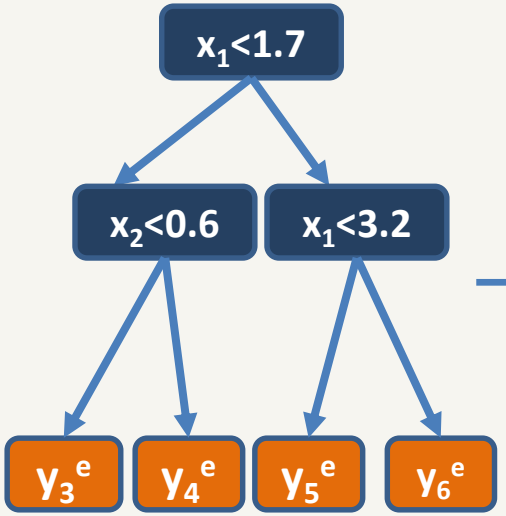
Y_1^e = media lui Y_1, Y_2, Y_3
Y_2^e = media lui $Y_{4:7}$

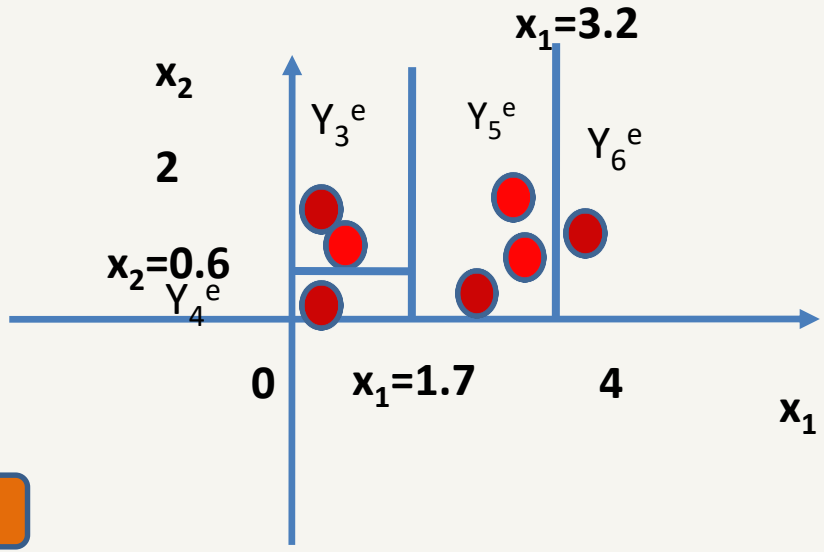




$$MSE = \frac{1}{7}\left(\sum_{i=1}^{3}\left(y_i - y_1^e\right)^2 + \sum_{i=4}^{7}\left(y_i - y_2^e\right)^2\right)$$

Arbore de regresie - exemplu





$$MSE = \frac{1}{7}\left(\sum_{i=N_1}^{N_2-1}\left(y_i - y_3^e\right)^2 + \sum_{i=N_2}^{N_3-1}\left(y_i - y_4^e\right)^2 + \sum_{i=N_3}^{N_4-1}\left(y_i - y_5^e\right)^2 + \sum_{i=N_4}^{N}\left(y_i - y_6^e\right)^2 \right)$$

Arbore de regresie - exemplu

- Când ne oprim ?
  - Când eroarea e mai mică decât un prag MSE < ?
    - Supraînvățare vs . Generalizare
  - O adâncime maximă a arborelui
    - Supraînvățare vs . Generalizare

Random Forest

Ansamblu de arbori

- Folosim mai mulți arbori
  - Foarte puternic
- Bootstrapping:
  - Se ia un subset F= ?% din setul de antrenament și se construiește un arbore
  - Eșantionare cu înlocuire
  - Repetă pentru N arbori

Random forest

- Ansamblu de arbori

SPLIT (Greedy):
MinGINI = RealMAX
For **each** dimension d = $x_1 \ldots x_N$
  For val = min($d_1 \ldots d_{N-1}$): max($d_1 \ldots d_{N-1}$)
    Split between val_{d_i} and val_{d_i+1}
    Subset value = the majority of values
    Compute GINI.
      If less than MinGINI, store
  end
End
Use dimension and val that lead to MinGINI

Ansamblu de arbori standard

SPLIT (Greedy):
MinGINI = RealMAX
For **randomly selected N_1** dimensions from $x_1 \ldots x_N$
  For val = min($d_1 \ldots d_{N-1}$): max($d_1 \ldots d_{N-1}$)
    Split between val_{d_i} and val_{d_i+1}
    Subset value = the majority of values
    Compute GINI.
      If less than MinGINI, store
  end
End
Use the dimension and val that lead to MinGINI

Random Forest

Rezultate cu Random forest