Machine learning

Decision Trees & $\boxed{\text{Ensemble methods}}$

xgboost

Wouter Gevaert & Marie Dewitte

Inhoud

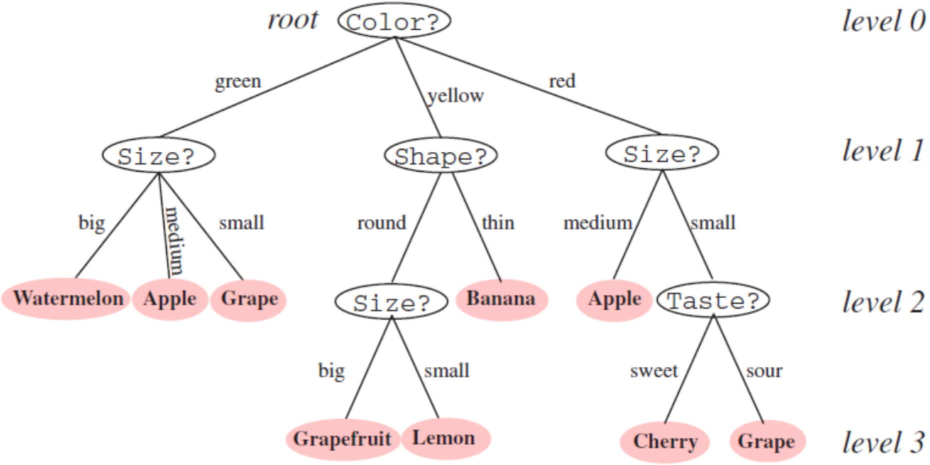
Decision trees voor classificatie

Decision trees voor regressie

Ensemble learning

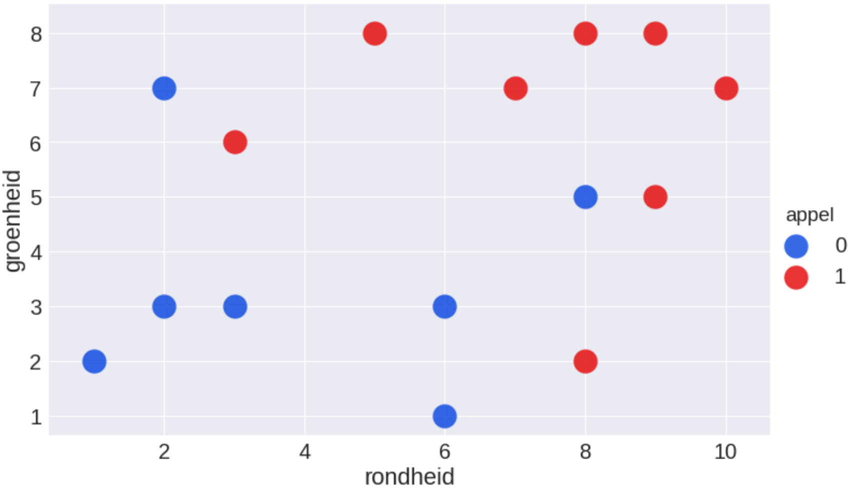
Decision trees voor classificatie

Introductie aan de hand van een voorbeeld

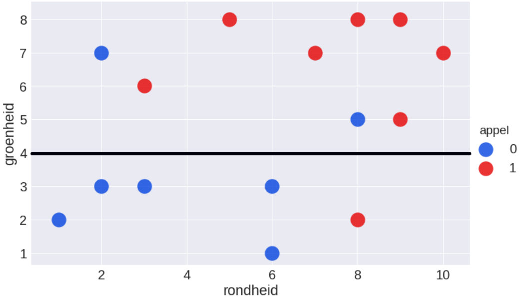


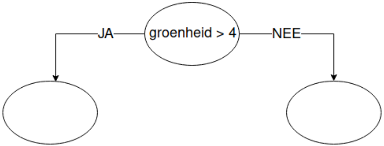
bron: `http://www.cse.msu.edu/~cse802/DecisionTrees.pdf`

Introductievoorbeeld - appels

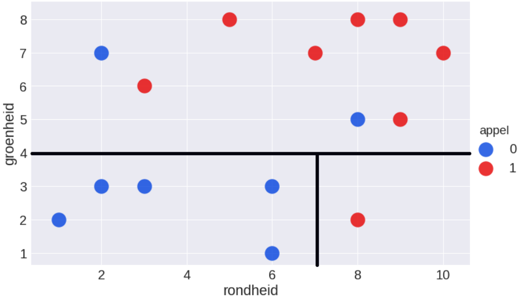


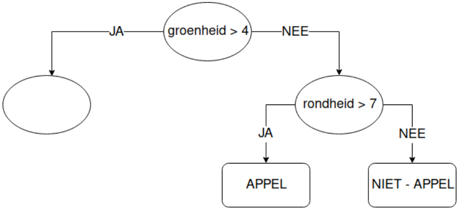
Introductievoorbeeld - appels



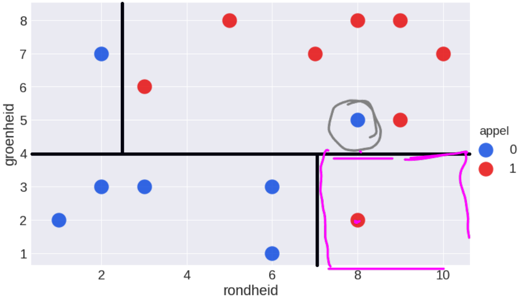


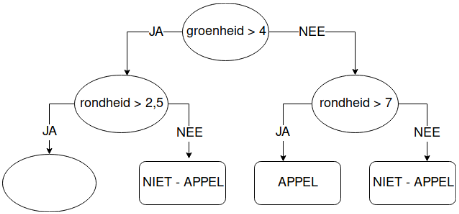
Introductievoorbeeld - appels



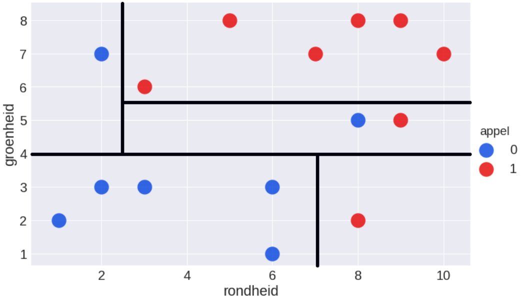


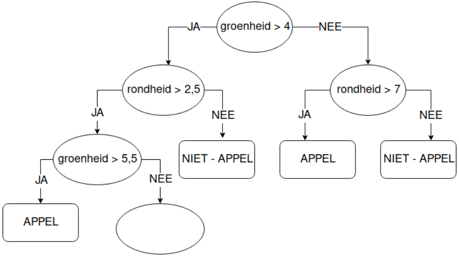
Introductievoorbeeld - appels





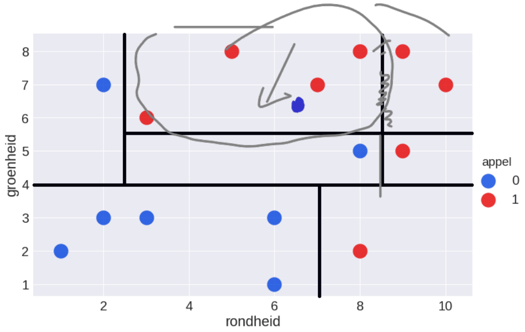
Introductievoorbeeld - appels

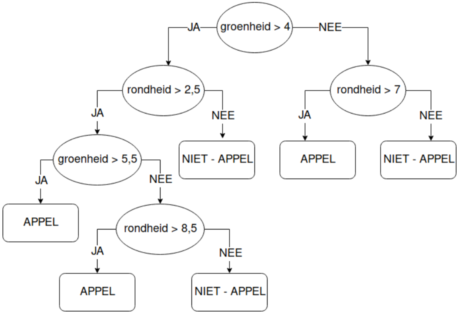




Introductievoorbeeld - appels

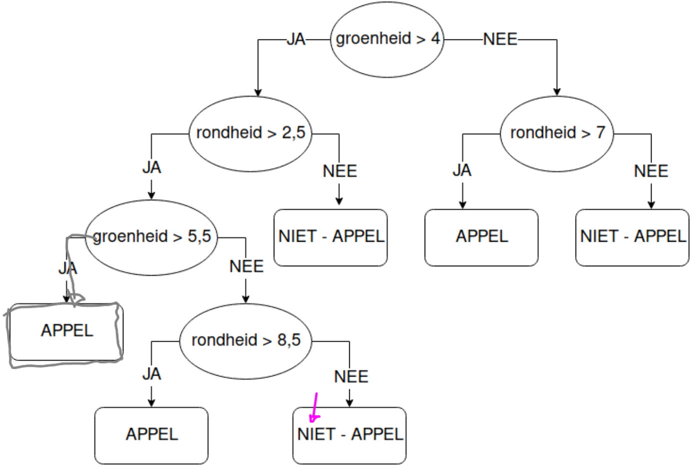
$\rightarrow \begin{cases} R = 6 \\ G = 7 \end{cases}$





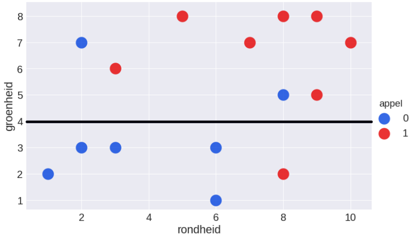
Introductievoorbeeld - appels

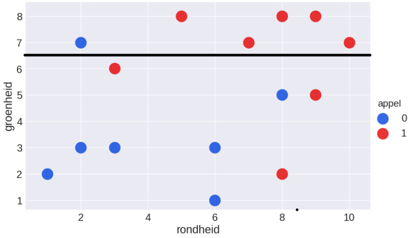
Classificeer een stuk fruit met een rondheid van 6 en een groenheid van 7

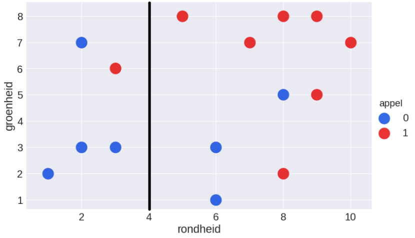


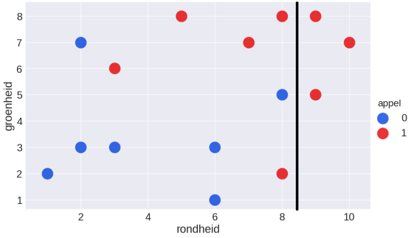
Splitsen van de boom

Hoe bepalen waar de boom te splitsen?









Splitsen van de boom

Entropy = mate van wanorde

$\boxed{\text{Entropy}}$ en information gain

Een goede split is deze waarbij de wanorde afneemt en beide kanten 'zuiverder (pure) worden'

Entropy H: maat voor de wanorde (gemiddelde informatieinhoud)

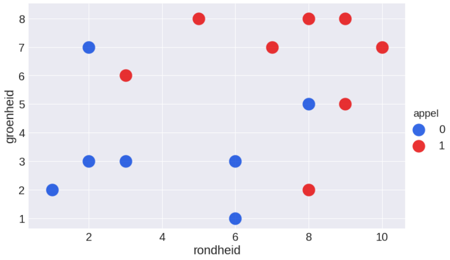
$$\underline{H} = \sum_{i=1}^{N} p_i \log_2 \left(\frac{1}{p_i} \right)$$

Entropy

$$H = -\sum_{i=1}^{N} p_i \log_2 (p_i)$$

Splitsen van de boom

Entropy en information gain



$$\text{Kans op een appel: } \frac{8}{15} = 0,53$$

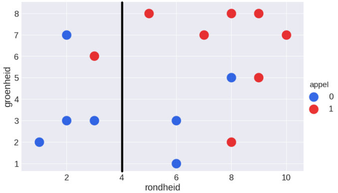
$$\text{Kans op een niet-appel: } \frac{7}{15} = 0,47$$

$$H = -(0,53 \times \log_2(0,53) + 0,47 \times \log_2(0,47)) = -(-0,99740) = \boxed{0,9974}$$

Splitsen van de boom

Entropy en information gain

Information gain = entropy voor de split - entropy na de split



Entropy links

$$p(appel) = \frac{1}{5} = 0,20$$

$$p(niet - appel) = \frac{4}{5} = 0,80$$

$$H = -(0,20 \log_2(0,20) + 0,80 \log_2(0,80)) = \boxed{0,72}$$

Entropy rechts

$$p(appel) = \frac{7}{10} = 0,70$$

$$p(niet - appel) = \frac{3}{10} = 0,30$$

$$H = -(0,70 \log_2(0,70) + 0,30 \log_2(0,30)) = \boxed{0,88}$$

Information gain = $0,9974 - (\frac{5}{15} \times 0,72 + \frac{10}{15} \times 0,88) = \boxed{0,17073}$

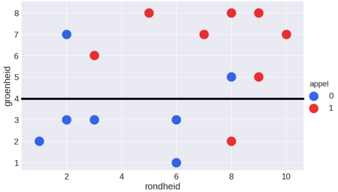
0,82

↳ inf. gain

Splitsen van de boom

Entropy en information gain

Information gain = entropy voor de split - entropy na de split



Entropy boven

$$p(appel) = \frac{7}{9} = 0,78$$

$$p(niet - appel) = \frac{2}{9} = 0,22$$

$$H = -(0,78 \log_2(0,78) + 0,22 \log_2(0,22)) = 0,76$$

Entropy onder

$$p(appel) = \frac{1}{6} = 0,17$$

$$p(niet - appel) = \frac{5}{6} = 0,0,83$$

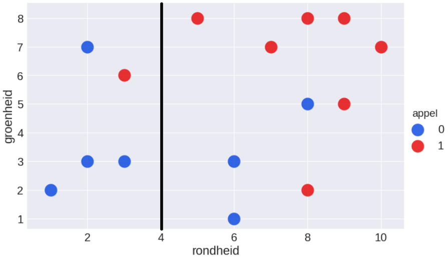
$$H = -(0,17 \log_2(0,17) + 0,83 \log_2(0,83)) = 0,66$$

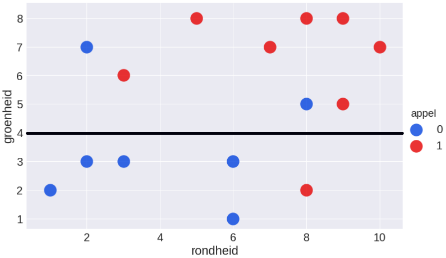
Information gain = $0,9974 - (\frac{9}{15} \times 0,76 + \frac{6}{15} \times 0,66) = \boxed{0,27740} \approx 0,28$

Splitsen van de boom

Entropy en information gain

Splits telkens bij de split die je de **hoogte information gain** oplevert.





Information gain = = 0,17073

Information gain = 0,27740

Splitsen van de boom

Gini impurity

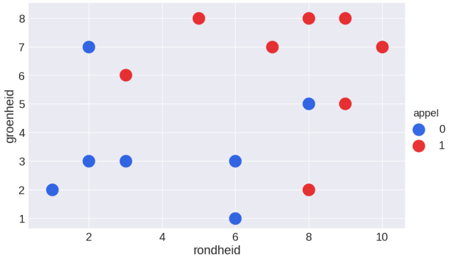
Alternatief voor het gebruik van de entropy.

De Gini impurity is een maat voor hoe vaak een willekeurig gekozen element van de set verkeerd gelabeld zou worden als het willekeurig werd gelabeld volgens de distributie van de labels in de subset.

$$G = 1 - \sum_{i=1}^{N} p_i^2$$

Splitsen van de boom

Gini impurity



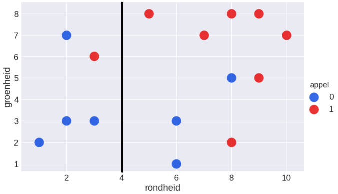
$$\text{Kans op een appel: } \frac{8}{15} = 0,53$$

$$\text{Kans op een niet-appel: } \frac{7}{15} = 0,47$$

$$G = 1 - (0,53^2 + 0,47^2) = \boxed{0,4982}$$

Splitsen van de boom

Gini impurity



Gini impurity links

$$p(appel) = \frac{1}{5} = 0,20$$

$$p(niet - appel) = \frac{4}{5} = 0,80$$

$$G = 1 - (0,20^2 + 0,80^2) = \boxed{0,32}$$

Gini impurity rechts

$$p(appel) = \frac{7}{10} = 0,70$$

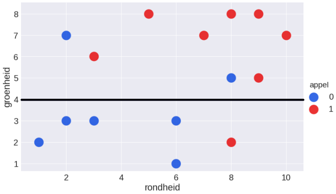
$$p(niet - appel) = \frac{3}{10} = 0,30$$

$$G = 1 - (0,70^2 + 0,30^2) = \boxed{0,42}$$

Gewogen Gini impurity = $\frac{5}{15} \times 0,32 + \frac{10}{15} \times 0,42 = \boxed{0,39}$

Splitsen van de boom

→ **Gini impurity** → default.



Gini impurity boven

$$p(appel) = \frac{7}{9} = 0,78$$

$$p(niet - appel) = \frac{2}{9} = 0,22$$

$$G = 1 - (0,78^2 + 0,22^2) = 0,34$$

0,39

Gini impurity onder

$$p(appel) = \frac{1}{6} = 0,17$$

$$p(niet - appel) = \frac{5}{6} = 0,0,83$$

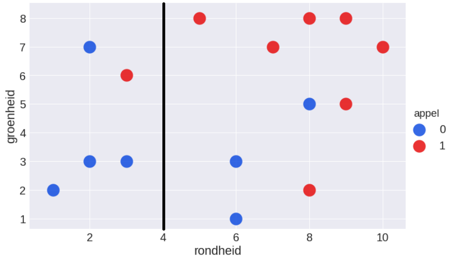
$$G = 1 - (0,17^2 + 0,83^2) = 0,28$$

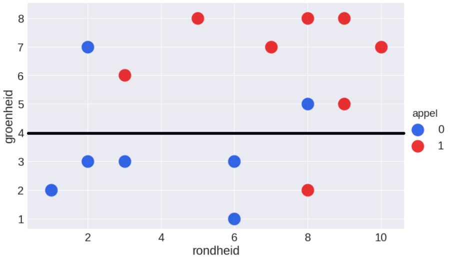
Gewogen Gini impurity = $\frac{9}{15} \times 0,34 + \frac{6}{15} \times 0,28 = \boxed{0,32}$

Splitsen van de boom

Gini impurity

Splits telkens bij de split die je de **laagste Gini impurity** oplevert.





Gewogen Gini impurity = = 0,39

Gewogen Gini impurity = 0,32

→ datasets met niet-lineariteit
→ DT

→ lineair dataset → lineaire regressie

Decision trees voor regressie

→ niet-geschaalde data

→ performantie

→ Transparant

$$\frac{25 + 30 + 35 + 38 + 48}{5}$$

$= \dots\dots$

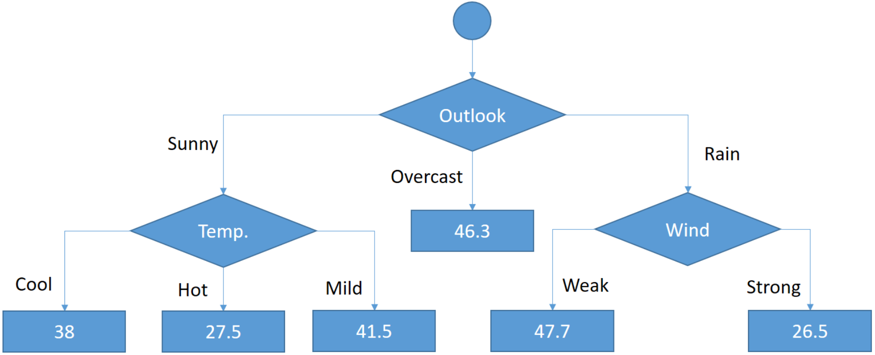
Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44
14	Rain	Mild	High	Strong	30

$$VAR = \frac{\sum (x_i - \bar{x})^2}{5}$$

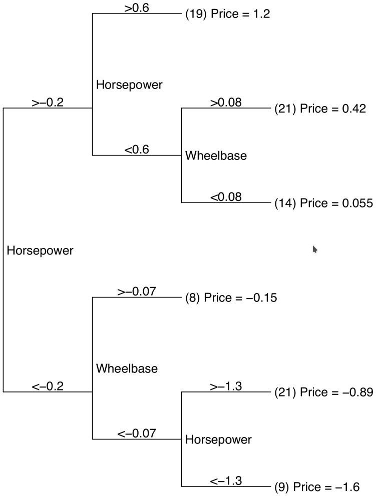
$\rightarrow$ VAR

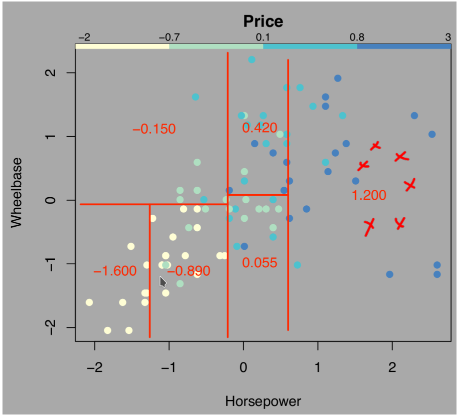
$\rightarrow$ VAR

Voorbeeld


Outlook
Sunny
Overcast
Rain
Temp.
46.3
Wind
Cool
Hot
Mild
Weak
Strong
38
27.5
41.5
47.7
26.5

Voorbeeld


>0.6
(19) Price = 1.2
Horsepower
>−0.2
>0.08
(21) Price = 0.42
<0.6
Wheelbase
<0.08
(14) Price = 0.055
Horsepower
>−0.07
(8) Price = −0.15
Wheelbase
<−0.2
>−1.3
(21) Price = −0.89
<−0.07
Horsepower
<−1.3
(9) Price = −1.6


Price
−2
−0.7
0.1
0.8
3
−0.150
0.420
1.200
−1.600
−0.890
0.055
Wheelbase
2
1
0
−1
−2
−2
−1
0
1
2
Horsepower

- Bagging
- Boosting
- Stacking

Ensemble learning

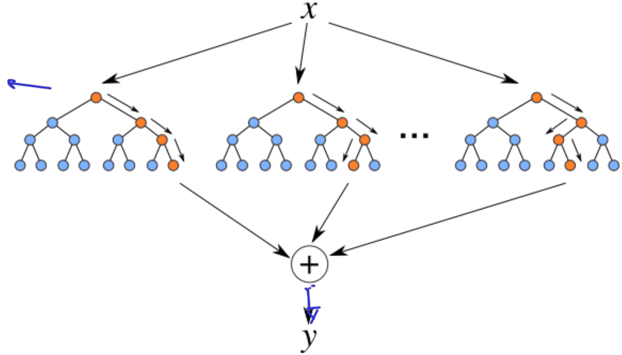
Ensemble learning

Problematiek van decision trees

Decision trees hebben de neiging tot overfitting.

Oplossing: **combineer** de voorspellingen van verschillende gerandomiseerde modellen (bijvoorbeeld decision trees) tot één model

weak classifier



Ensemble learning

JA = 60%.

NEE = 40%.

Given a jury of voters and assuming independent errors. If the probability of each single person in the jury of being correct is above 50% then the probability of the jury being correct tends to 100% as the number of persons increase.

Nicolas de Condorcet (1743 - 1794)

Ensemble learning

Overzicht Ensemble learning methodes

- Bootstrap aggregating (bagging)
- Boosting
- ~~Bayes~~ optimal classifier
- ~~Bayesian~~ model averaging/combination
- ~~Bucket~~ of models
- Stacking

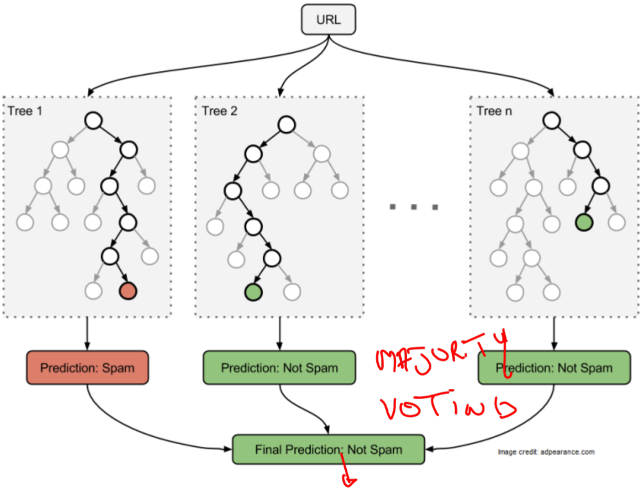
$\boxed{p(y \mid x)} = - \; - \; - \; .$

1)

Bagging

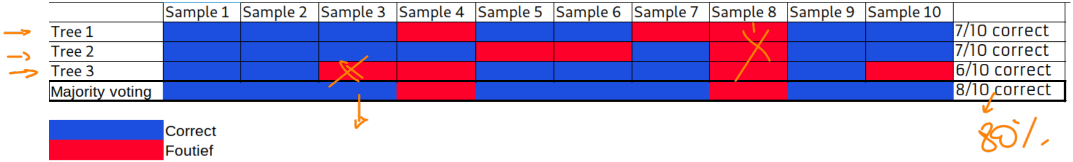
Random Forest Trees
(Bagging)

Concept van bagging - Majority voting



Bagging

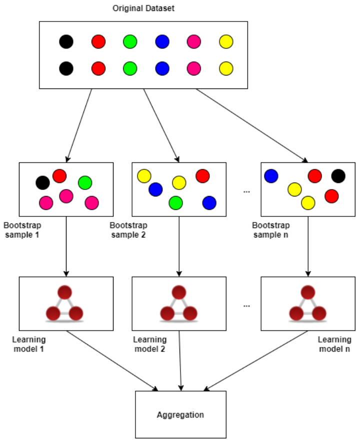
Concept van bagging - Majority voting



Bagging

De techniek van bagging

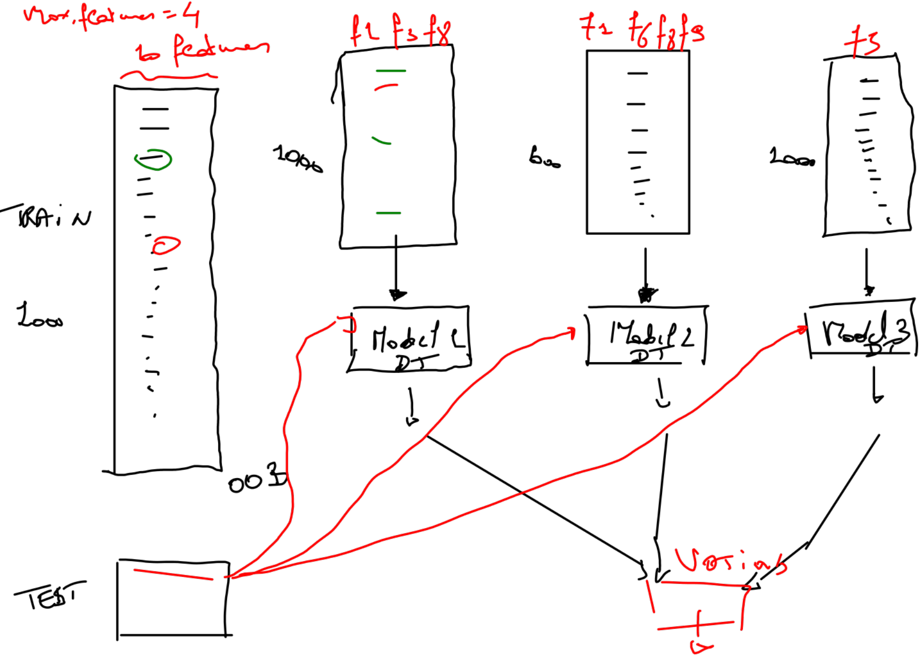
Methode om de variantie te reduceren en overfitting te vermijden.



Kies verschillende subsets (bags) uit de trainingset.

Willekeurige selectie met teruglegging

Train op elke subset een classifier



BAGGING is GOED BIJ UITSCHIETER

Bagging

Random Forest Trees

- Gini
- ENTROPY

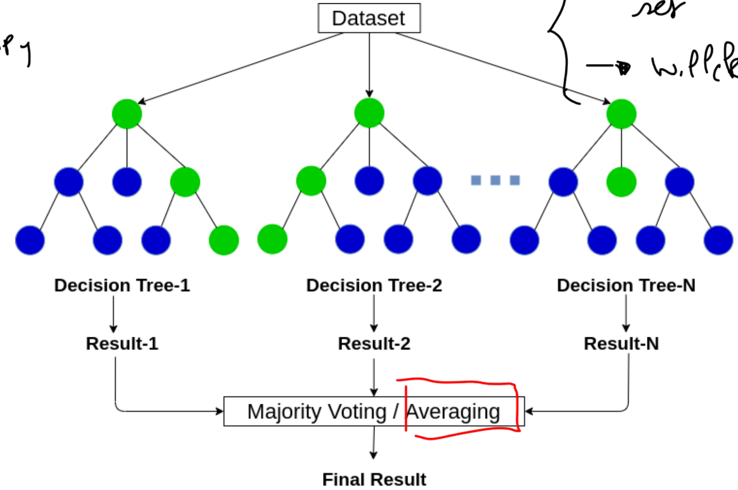
→ REKENKRACHT

→ EXTRATREES

Extremely Randomized Trees.

→ volledige key set

→ willekeurig splits.



Bagging

Random Forest Trees - hyperparameters

n_estimators: number of trees in the forest. Meestal hoe meer hoe beter.

Criterion: Gini of Entropy (default Gini)

Maximum number of features: het maximum aantal features per boom.

- int : aantal features
- float: percentage
- 'auto': max_features = vierkantswortel van totaal aantal features
- 'sqrt': max_features = vierkantswortel van totaal aantal features
- 'log2': log van het totaal aantal features
- Default worden alle features gebruikt.

Random forest trees

Random Forest Trees - hyperparameters

max_depth: de maximale diepte van de boom. Als je te maken hebt met noisy data is het aan te raden de maximale diepte beperkt te houden.

min_samples_split: het minimum aantal samples nodig om binnen een boom te blijven splitsen.

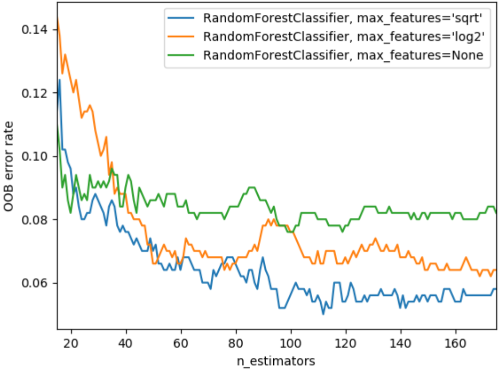
min_samples_leaf: het minimum aantal samples dat zich aan een blad van de boom moet bevinden. Hoe groter deze waarde, hoe minder vatbaar voor overfitting.

Bootstrap aggregating: Bagging (zie verder). Staat default op True.

Bagging

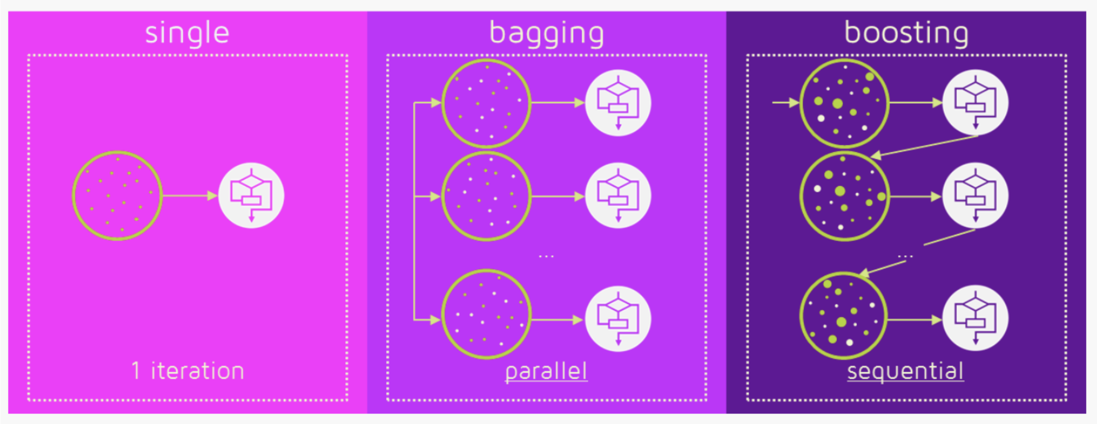
Random Forest Trees - hyperparameters

oob_score: de gemiddelde error bij het testen van een sample op bomen die niet op deze sample getraind zijn geweest.



Boosting

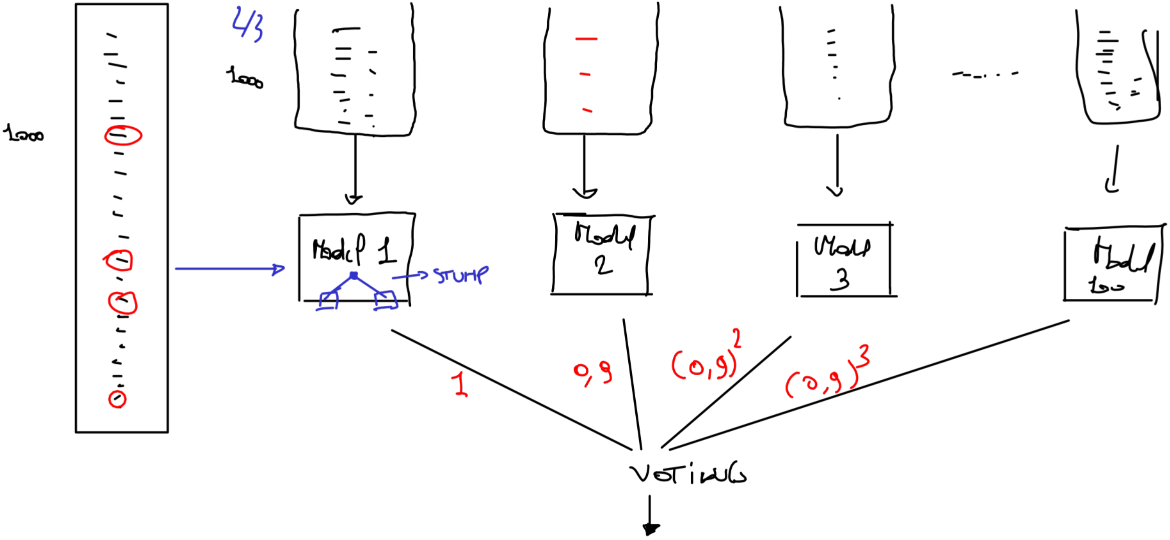
Principe



Adaboosting

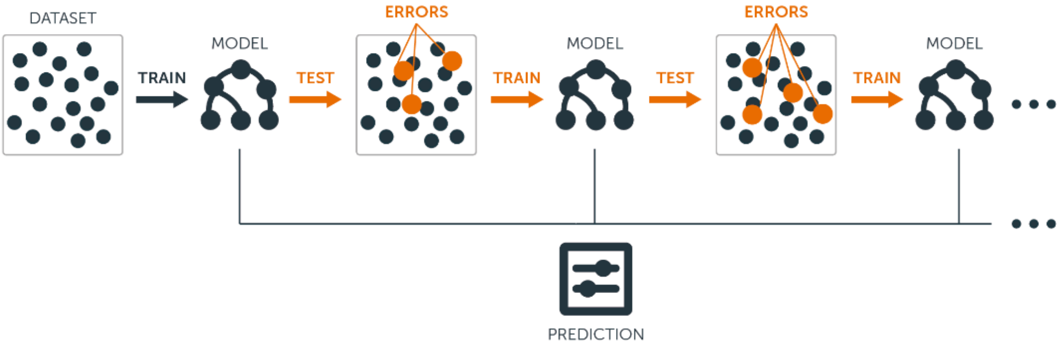
Learning rate = 0,9

Accuracy



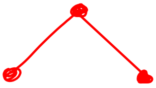
Boosting

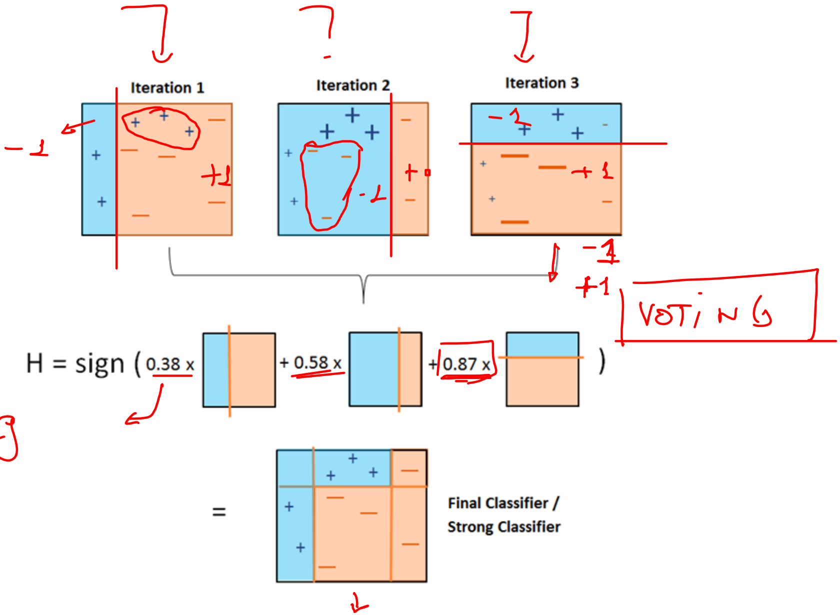
Adaboost



Boosting

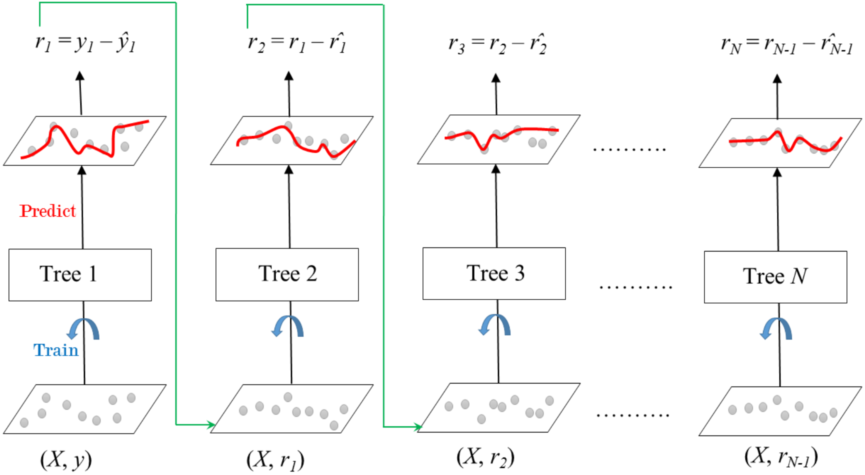
Adaboost



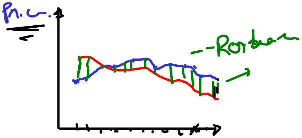


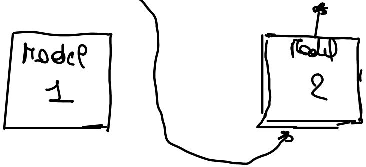
Boosting

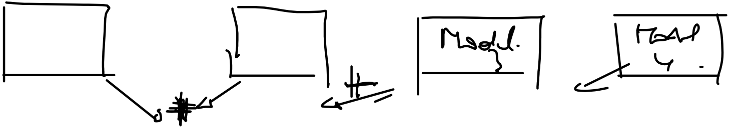
Gradient Boosting



Regresie







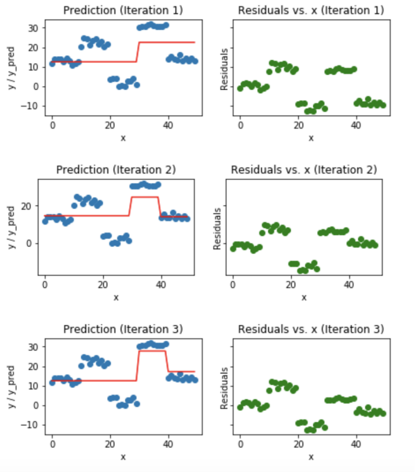
→ CLASSIFICATIE

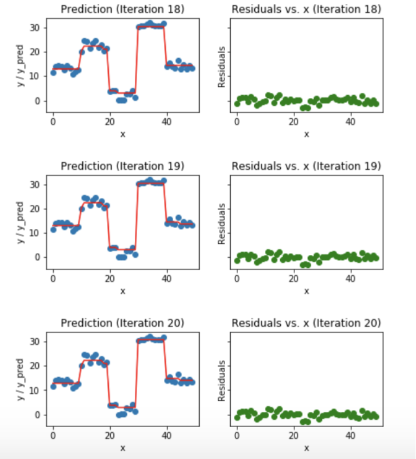
→ probabl

→ probabil.

Boosting

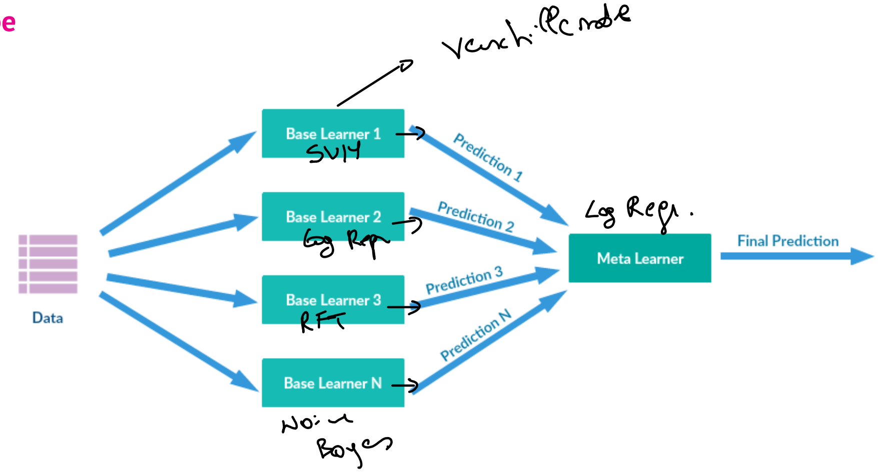
Gradient Boosting





Stacking

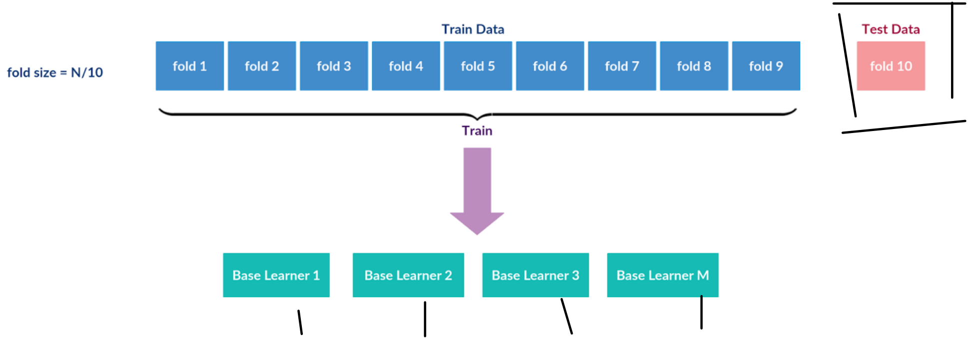
Principe



Stacking

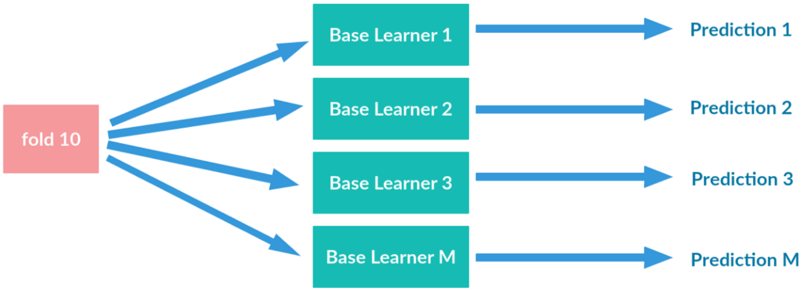
Principe

$K = 10$



Stacking

Principe



Stacking

Principe

Data Point #	prediction from base learner 1	prediction from base learner 2	prediction from base learner 3	prediction from base learner M	actual
1	$\hat{y_{11}}$	$\hat{y_{12}}$	$\hat{y_{13}}$	$\hat{y_{1M}}$	y_1
2	$\hat{y_{21}}$	$\hat{y_{22}}$	$\hat{y_{23}}$	$\hat{y_{2M}}$	y_2
...	...	...	...	...	...
N	$\hat{y_{N1}}$	$\hat{y_{N2}}$	$\hat{y_{N3}}$	$\hat{y_{NM}}$	y_N

Ensemble learning

Samenvattend overzicht

Bagging

- Reduceert de variantie
- Robuust tegen uitschieters en noisy data
- Dikwijls via Random Forest Trees

Boosting

- Verhoogt de accuraatheid
- Niet robuust tegen uitschieters of noisy data

Stacking

- Ensemble van strong learners
- Metalearner die de opimale combinatie leert van de base learners