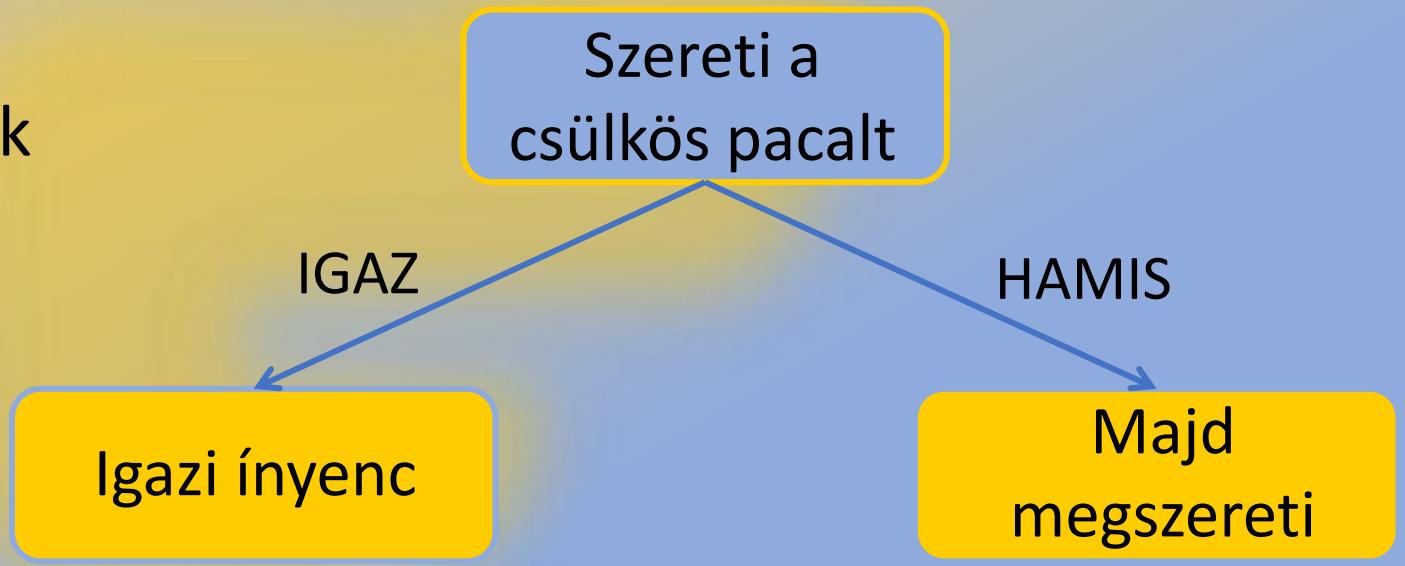
4. Előadás
Döntési Fák
A CART tanító algoritmus

A döntési fa helye az ML-ben

- Olyan, mint egy svájci bicska!
- Mindig kéznél van, mindenre jó, de szinte semmire sem a legalkalmasabb.
  - Ahogy a kézifűrész jobb, mint a bicskás fűrész.
  - Ahogy a séfkés jobb, mint a bicskás kés.
  - Ahogy a harapófogó jobb, mint a bicskás fogó.
  - De lehet-e egy séfkéssel anyacsavart meglazítani?
- A döntési fák különösen hasznosak **gyorsaság**uk és **egyszerűség**ük miatt adatfeltérképezésre, gyors eredmények megmutatására, változók közötti kapcsolatok megmutatására.

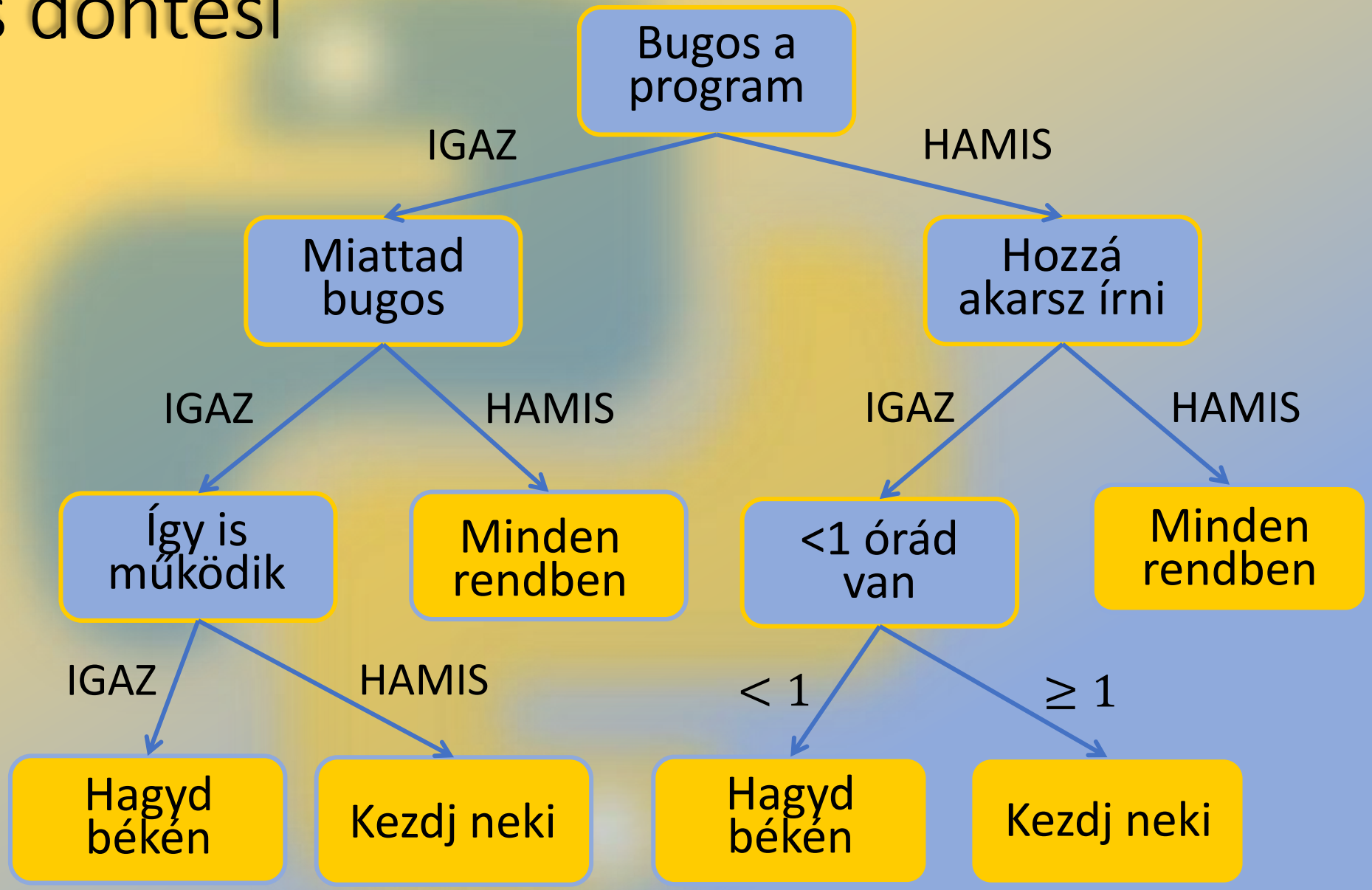
Az alap elképzelés

- A döntési fák sokoldalú gépi tanulási algoritmusok, amelyek mind bináris és multioutput osztályozást, illetve regressziót is képesek végrehajtani. Könnyen illeszthetők komplex adathalmazokra. Ez az erősségük és gyengeségük is egyben.
- Az algoritmus alapja, hogy mintaegyedeket **osztályoz** változóikban felvett értékeik alapján.
- A balra látható döntési fa egy **tönk**: nincs internális csomópontja.

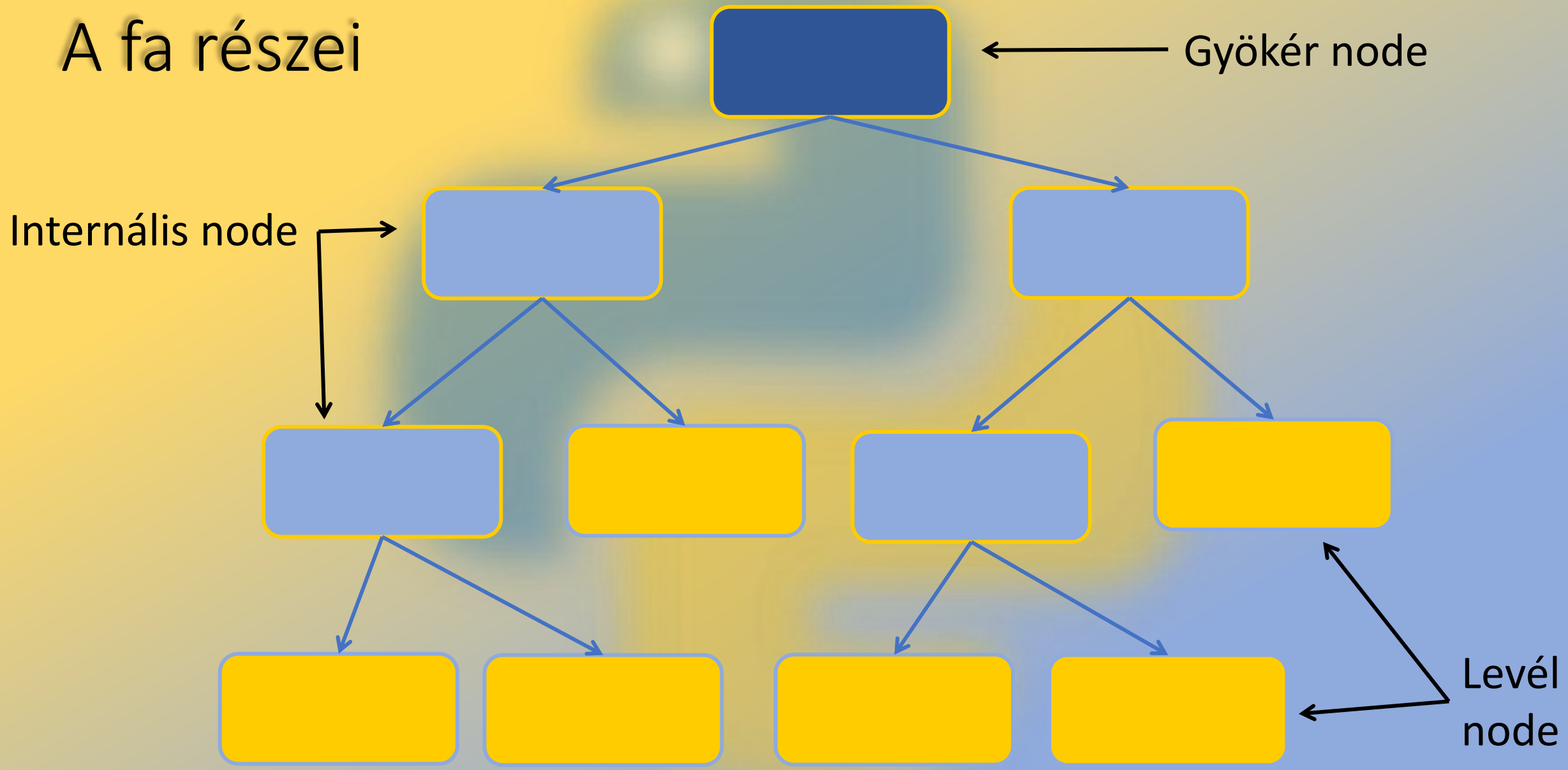


Egy hasznos döntési fa példának

- A végső osztályok típusa lehet diszjunkt osztály és folytonos változó is!
- A mintaegyedek a csomópontok kérdéseire válaszolnak, változóik alapján.



A fa részei



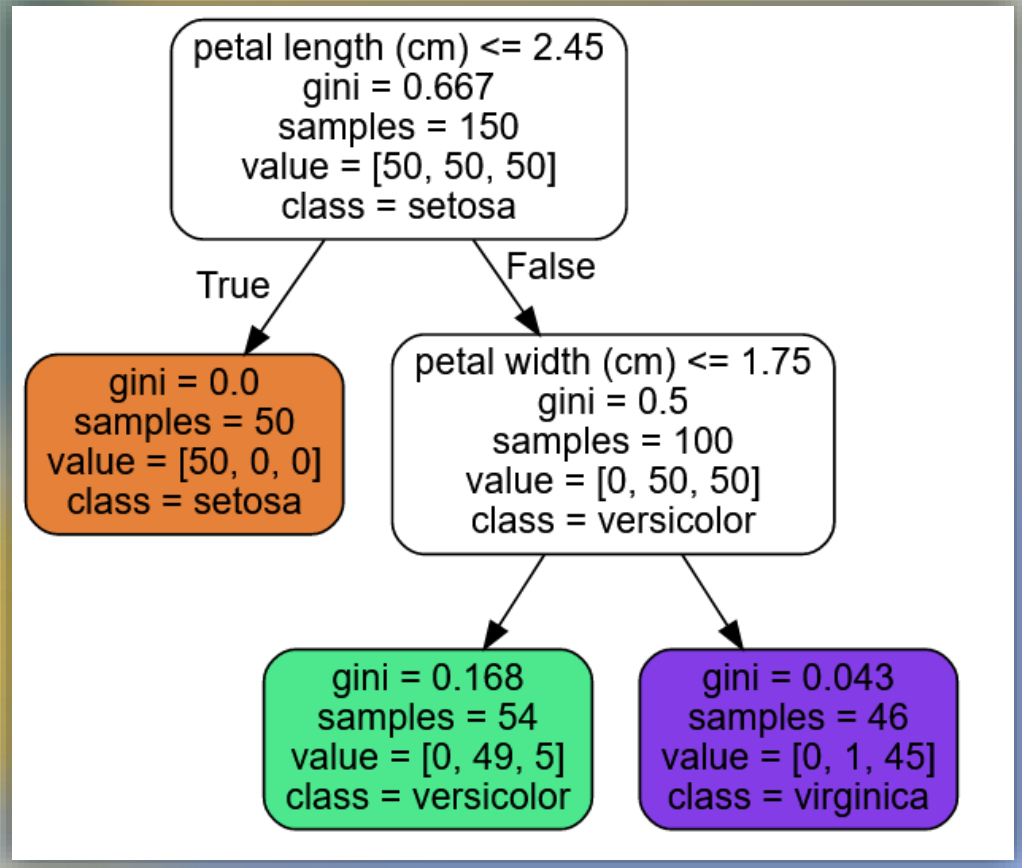
Egy kezdeti döntési fa az Írisz adathalmazon

```python
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

iris = load_iris()
X = iris.data[:, 2:] # szirom hossz és szélesség
y = iris.target

tree_clf = DecisionTreeClassifier(max_depth=2, random_state=42)
tree_clf.fit(X, y)
```

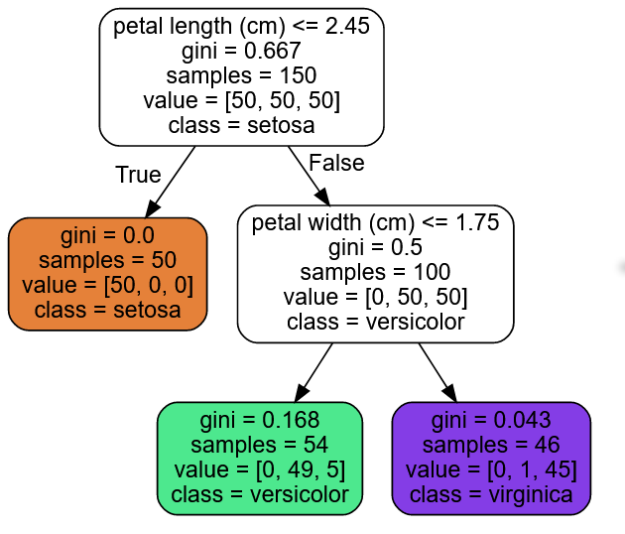
```
DecisionTreeClassifier(max_depth=2, random_state=42)
```

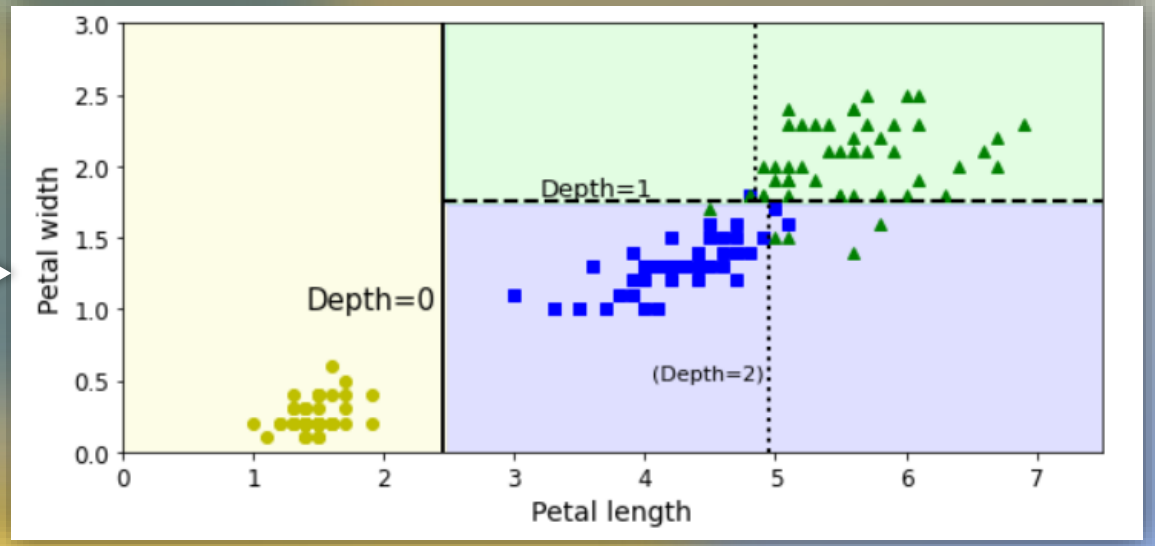


- Hogyan osztályoznánk be egy új virágot?
- A gyökércsomóponttól indulva (0. szint), mindig a node által feltett kérésre válaszolva, ameddig a mintaegyed el nem éri valamelyik **terminális régiót**.
- A gyökér kérdése pl. hogy a szirom hossz nagyobb-e mint $2.45cm$.

- Melyik lépés maradt ki a regresszor tanításából?
- Mi lehet a $gini = 0.043$?

A modell ábrázolása (**white box modell**)





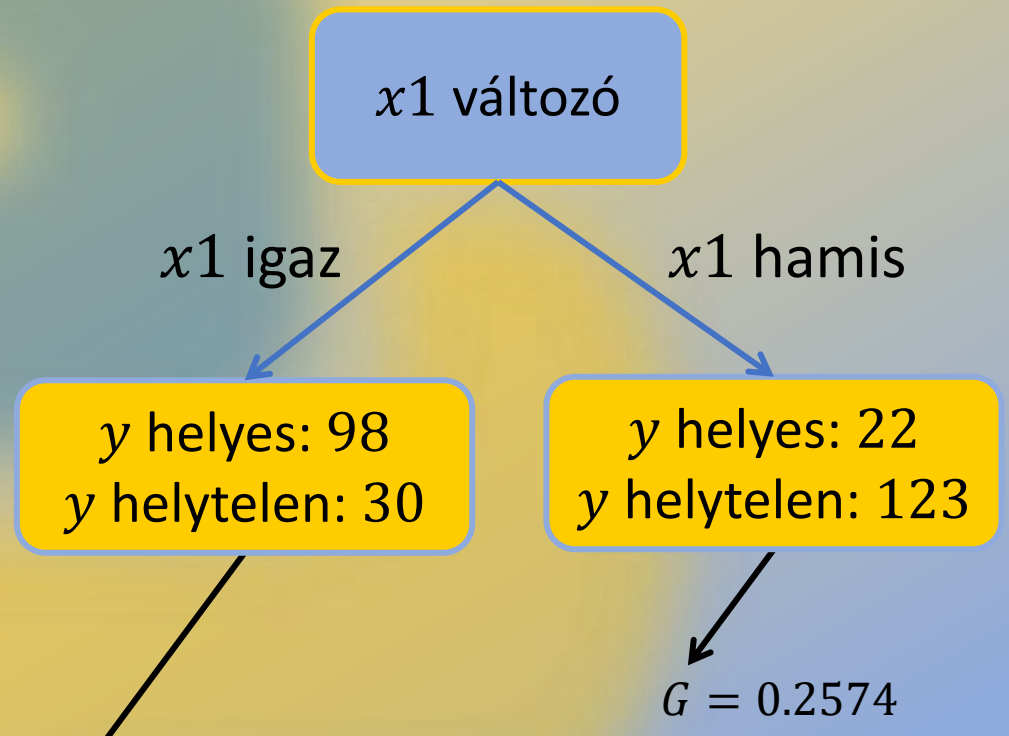
- Az ábra ennek a döntési fának a határvonalait mutatja. A vastag vonal a gyökérből származó határ. Mivel a bal oldali halmaz teljesen tiszta, nem lehet tovább bontani. De a jobb oldali részhalmaz továbbra is kevert, ezért a jobb oldali 1. szintű belső node tovább bontja $petal\ width = 1.75cm$-nél.
- Mivel a $\max\_depth = 2$, a modell itt megáll, de ha ez az érték 3 lenne, a függőleges pöttyözött vonal mentén történne a szétválasztás.

A levelek jóságának mérése (tisztaság)

- Azok a változók, amelyek nem tudják 1: 0 arányban szeparálni az egyedeket, tisztátalannak számítanak. Ennek egyik mérőszáma a **Gini**-index.

$$Gini = 1 - P(A)^2 - P(B)^2$$

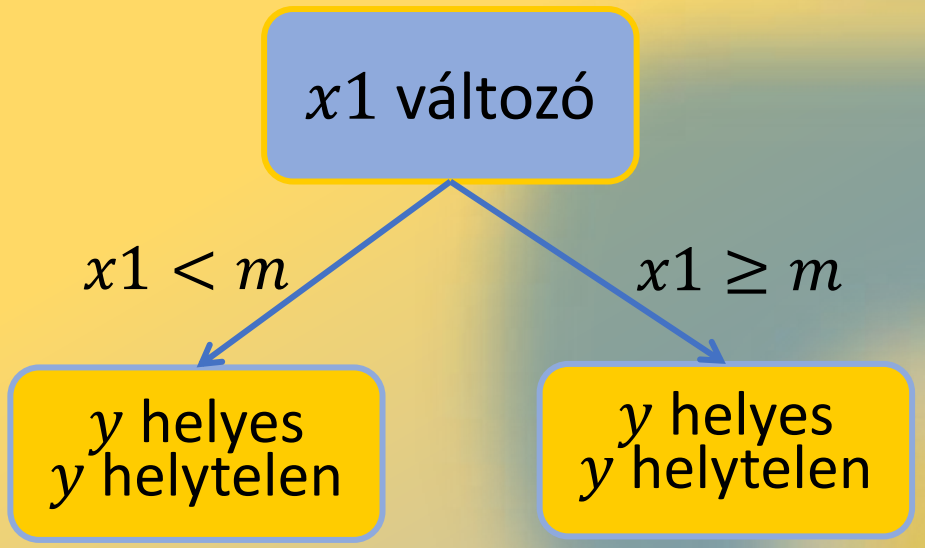
- Egy változó Gini-indexe levelei Gini-indexének súlyozott átlaga.



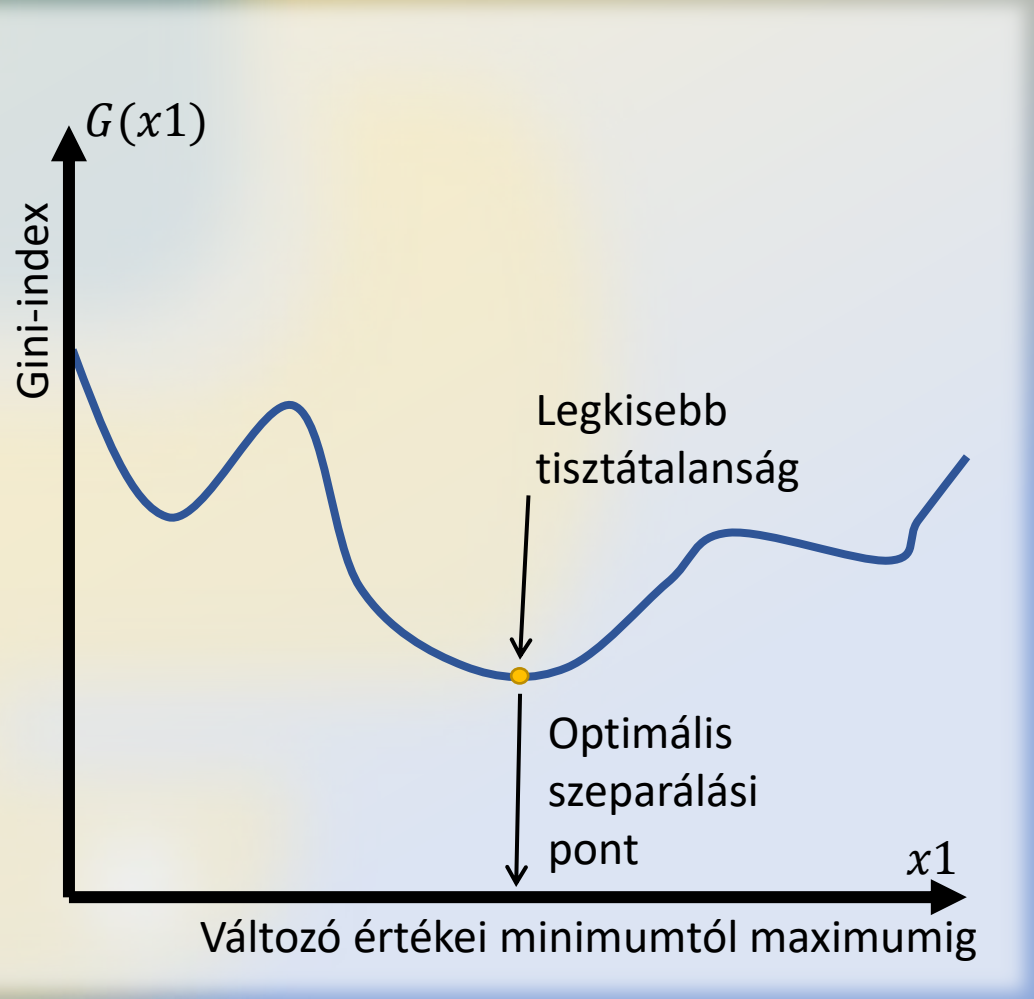
$$G = 1 - \left(\frac{98}{98+30}\right)^2 - \left(\frac{30}{98+30}\right)^2 = 0.3588$$

$$G(x1) = \left(\frac{128}{128+145}\right)0.3588 + \left(\frac{145}{128+145}\right)0.2574 = 0.3049$$

Szeparáció folytonos változó esetén

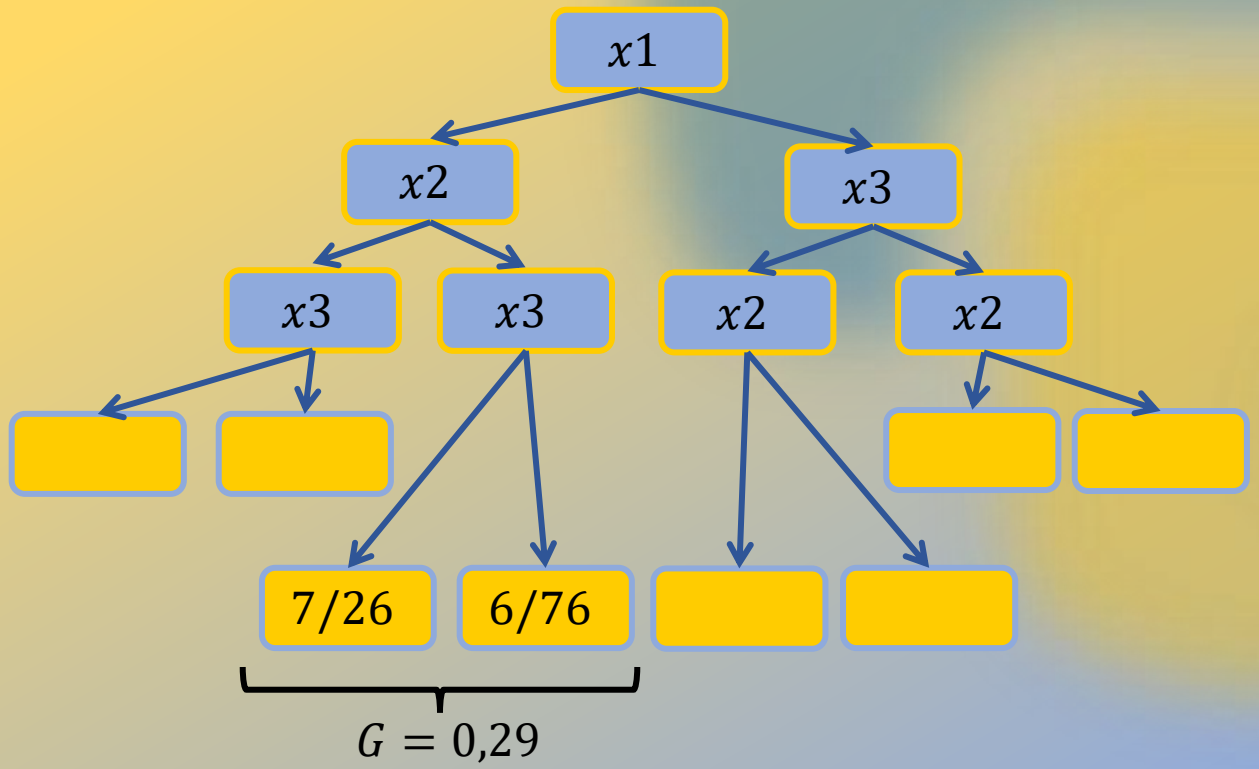


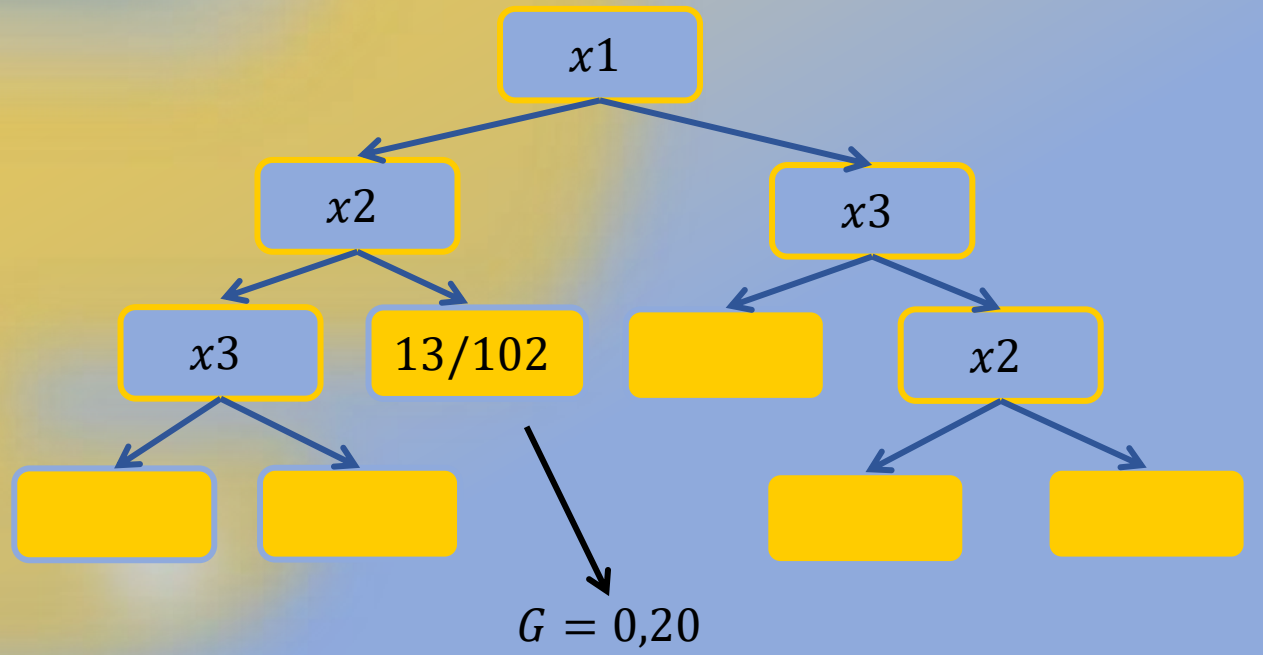
- A folytonos változónak minden értékéhez tartozik egy Gini érték.
- Ezt felírva kapjuk a Gini-függvényt adott változóra kiszámolva.



Mikor érdemes szeparálni?

- Amikor egy csomópontnak magasabb a tisztátalansága tovább bontáskor, felesleges a szeparáció, és levélcsomópont válik belőle.
- Gyökércsomópont abból a változóból válik, amelynek a legalacsonyabb a tisztátalansága.





A **CART** tanító algoritmus

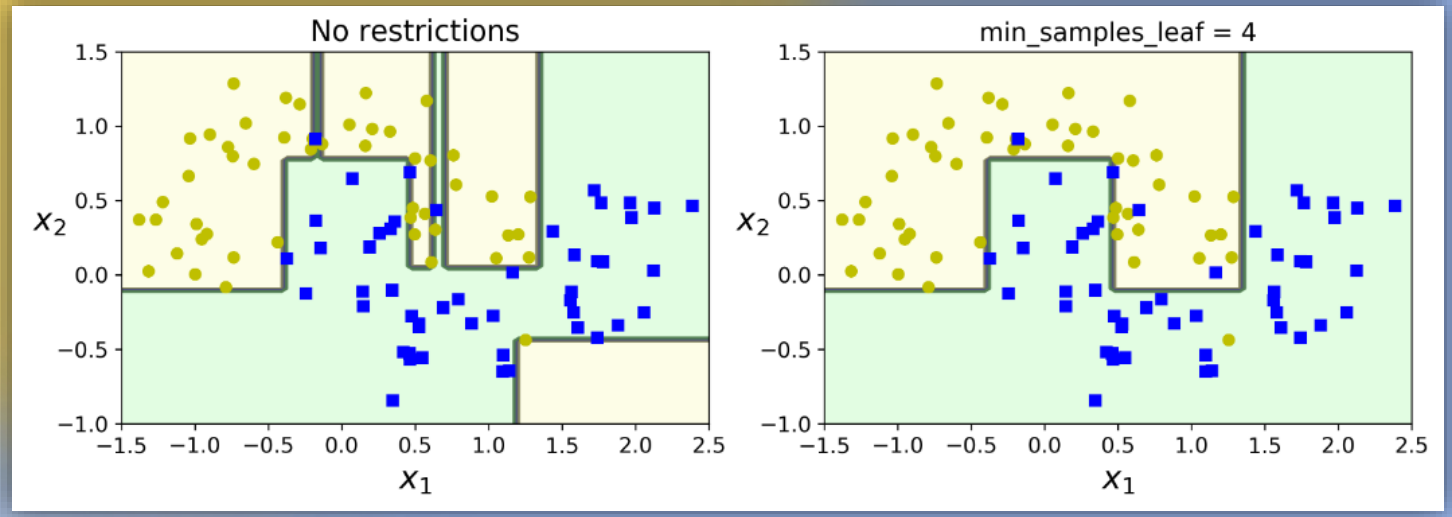
- A scikit-learn a Classification And Regression Trees algoritmust használja a növekvő fák tanításához.
- Az ötlet meglehetősen egyszerű: először az algoritmus a tanító pontokat k jellemző és t_k küszöbérték szerint kettéválasztja.
- Az algoritmus olyan (k, t_k) párokat keres, amelyekkel a létrejövő részhalmazoknak a lehető legalacsonyabb a tisztátlansága.
- Ezt addig ismétli rekurzívan, ameddig a szintek száma el nem éri a $\max\_depth$ hiperparamétert.
- A CART osztályozó költségfüggvénye:

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

- Ahol:
  - $G_{left/right}$: a bal/jobb adathalmaz tisztátalansága
  - $m_{left/right}$: az egyedek száma a bal/jobb halmazban
  - G: Gini-index

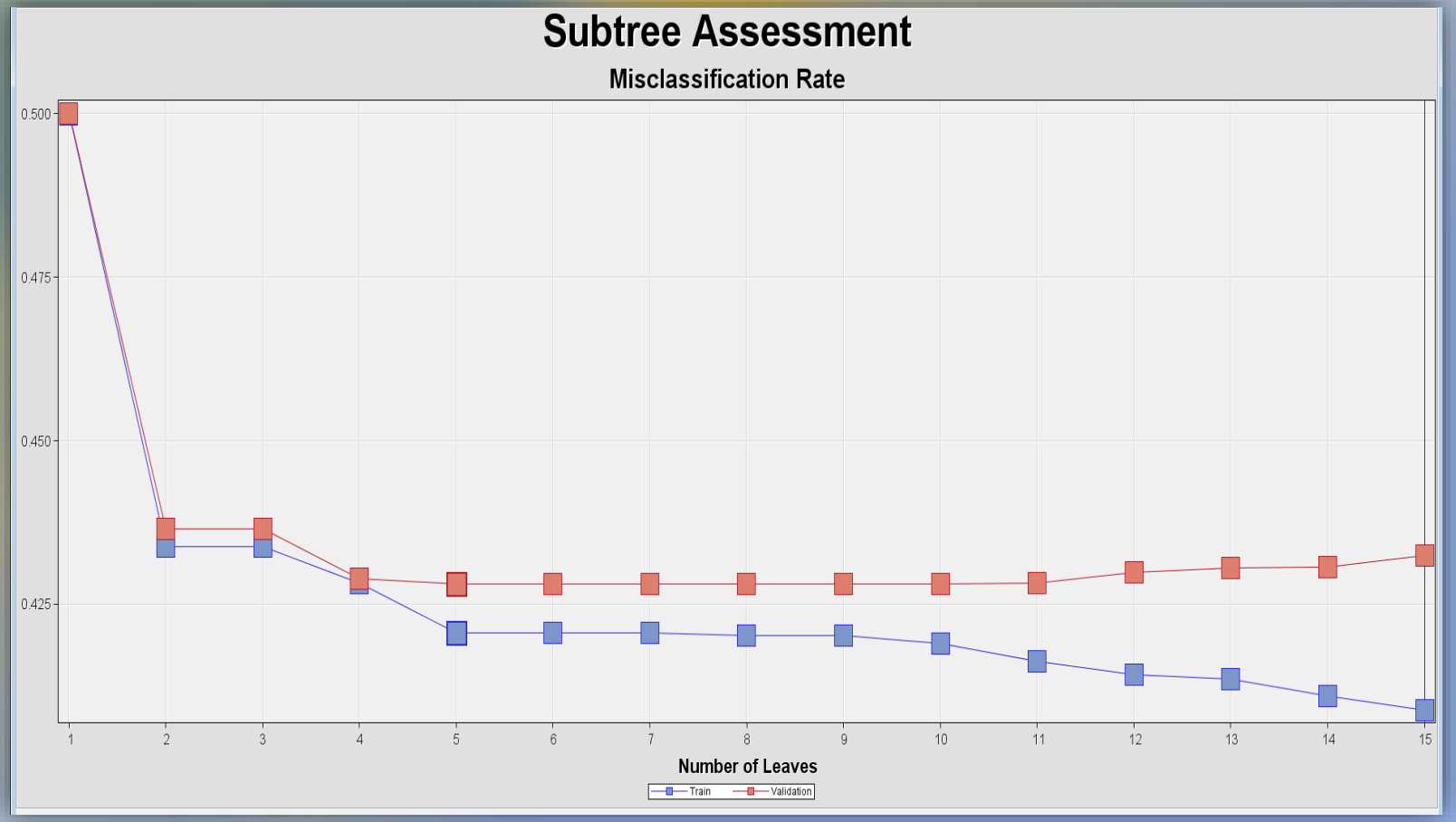
Regularizációs hiperparaméterek

- A döntési fák meglehetősen kevés előfeltételezéssel élnek az adatok irányába. Ha megkötések nélkül tanítjuk, akkor a fa struktúrája nagyon szorosan fog alkalmazkodni a tanító pontokhoz.
- Ahhoz, hogy a túltanulást elkerüljük, bizonyos megszorításokat kell tennünk a döntési fa illesztési szabadsága felé (**regularizáció**).
- Néhány hiperparaméter: hogyan kell állítani a max és min értékeket?
  - *Max_depth*: a döntési fa maximális mélysége (0-ról indul a gyökérrel!)
  - *Min_samples_split*: a minimum mintaegyedszám, ami ahhoz kell, hogy szeparáljon a csomópont
  - *Min_samples_leaf*: egy levélbe bekerülő minimális mintaegyedszám
  - *Max_leaf_nodes*: a levelek max. száma
  - *Max_features*: maximum változó amit ki kell értékelni szeparálás előtt

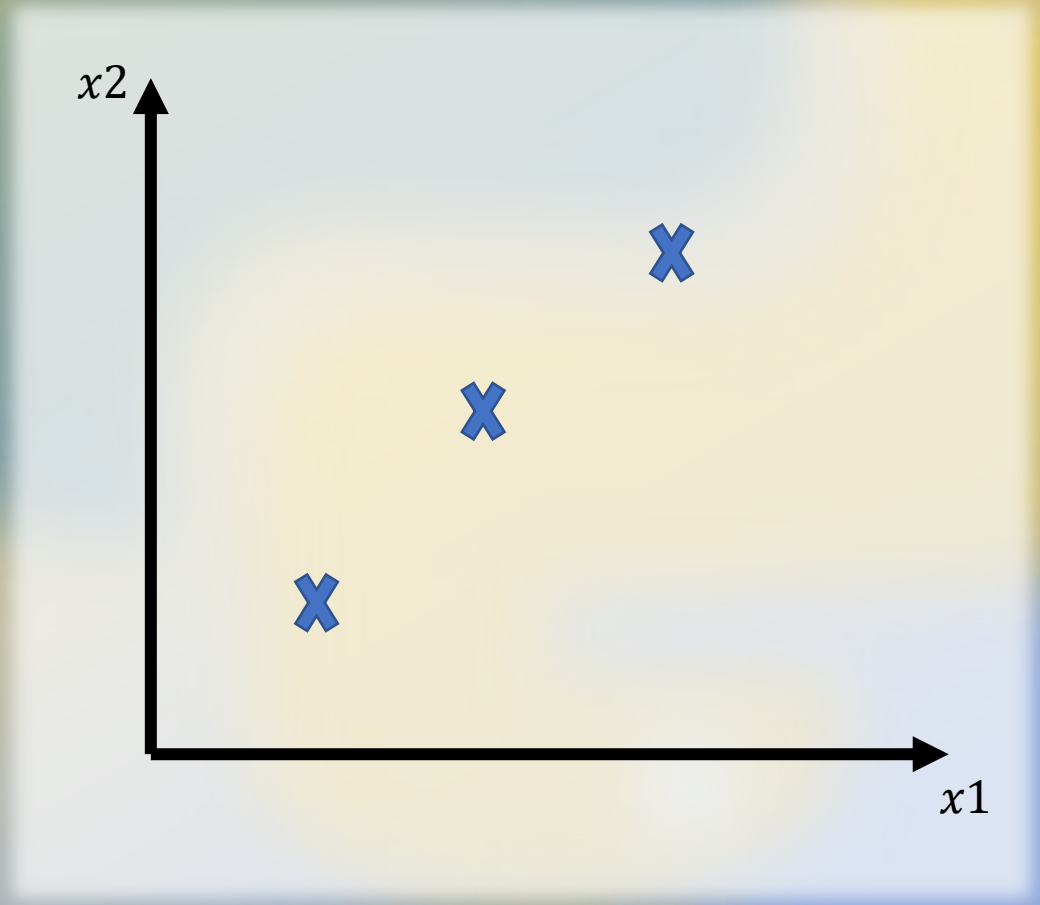


Félreosztályozási ráta: subtree assessment plot

- Hány változó szükséges a predikcióhoz?
- Csakúgy mint a többi modellnél, a döntési fáknál is megfigyelhető a túltanulás.
- Ebben az esetben a tanító hiba csökken, de a validációs hiba emelkedik.
- Annyi változót érdemes meghagyni, amennyinél a lehető legalacsonyabb a validációs hiba.
- Ezt ábrázolja a subtree assessment plot: hány levél kell a minimális validációs hiba eléréséhez?

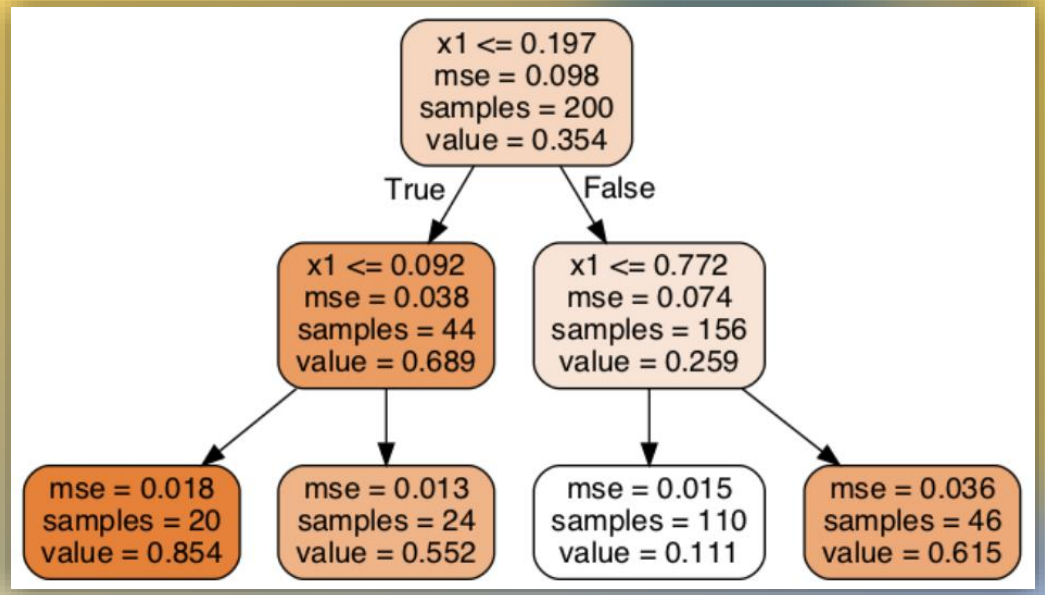


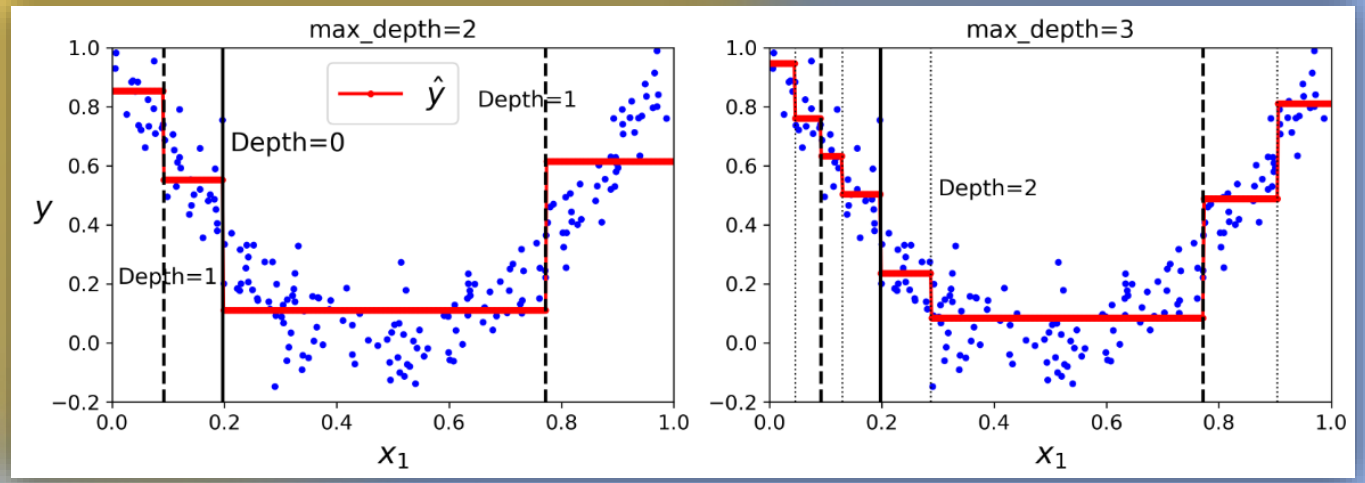
Feladat: hányféleképpen tudja egy **tönk** beosztályozni az alábbi pontokat, $x1$ és $x2$ változó szerint?



Regresszió döntési fákkal

- Tanítsunk zajos adatokon egy DecisionTreeRegressor-t. A létrejövő modell nagyon hasonlít az osztályozóhoz. Mi a levél jóságának mértéke?
- Az egyetlen különbség az, hogy a levelek értékeket reprezentálnak.
- Ebben az esetben a predikció egyszerűen az átlaga a terminális régióba bekerült mintaegyedek célváltozóikban felvett értékének.
- A modell mélysége 2, de látható, hogy nézne ki, ha 3 lenne.



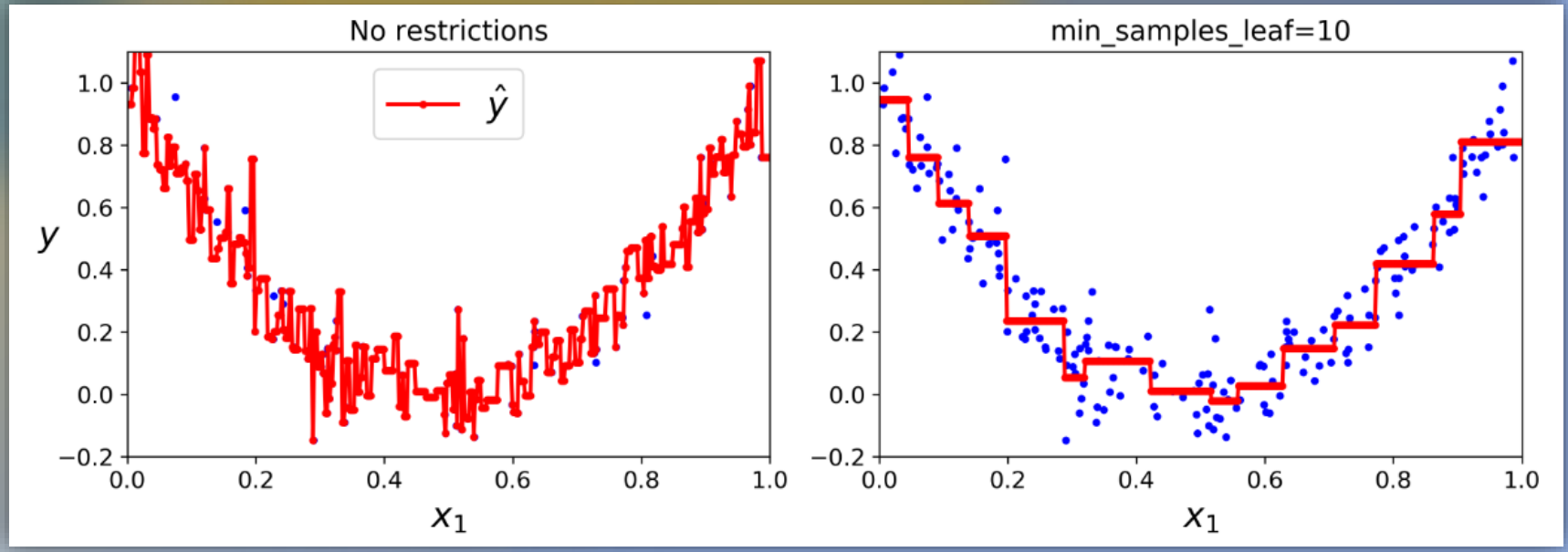


Tanítás regresszió esetén

- A regresszor fák a tisztátalanság minimalizálása helyett az MSE-t minimalizálják, így a CART regresszor költségfüggvény a következő:

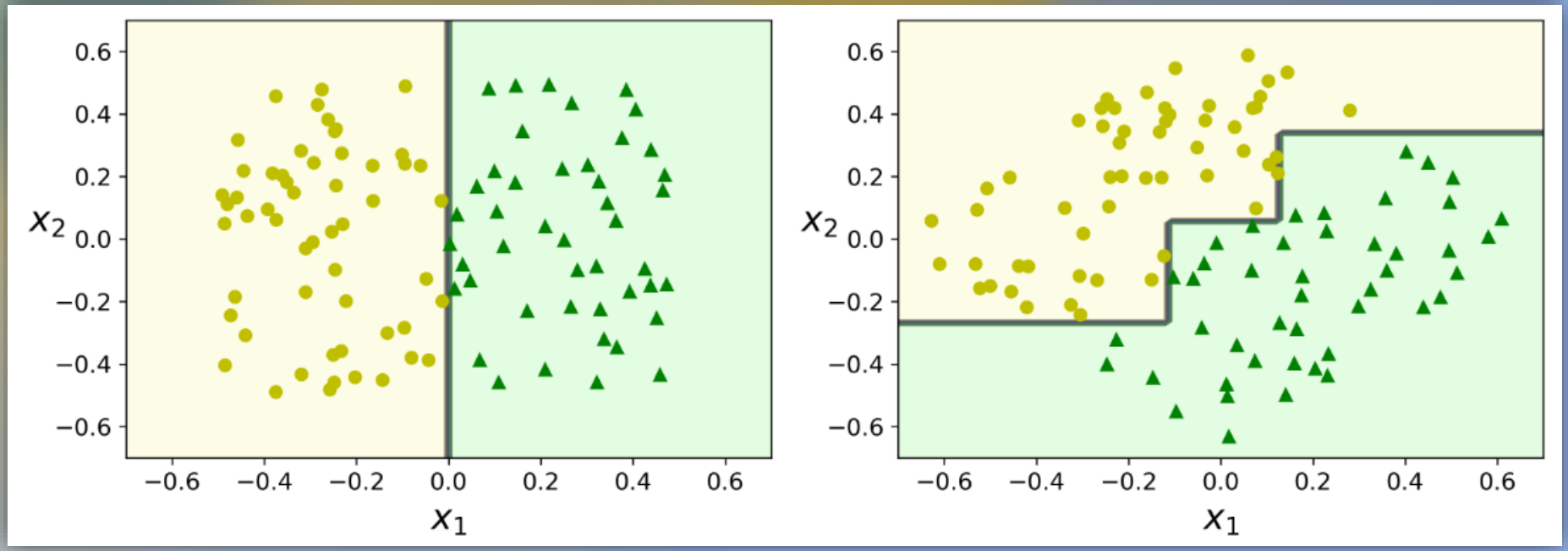
$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}}$$

- Ahol:
  - $MSE_{node} = \sum_{i \epsilon node}(\hat{y}_{node} - y^{(i)})^2$
  - $\hat{y}_{node} = \frac{1}{m_{node}}\sum_{i \epsilon node} y^{(i)}$
- A regressziós fák is nagyon hajlamosak túltanulásra.
- Regularizáció nélkül a bal, vele pedig a jobb oldali modellt kapjuk.



Instabilitás: adathalmaz rotációja

- Ahogy már biztosan észrevettük, a döntési fák ortogonális döntési határokkal dolgoznak (minden szeparáció a tengelyekre párhuzamosan történik).
- Ez a tulajdonságuk az adathalmaz rotációval szemben érzékennyé teszi őket.
- Az ábrán egy egyszerű, lineárisan szeparálható adathalmazt látunk.
- A bal oldalon egy döntési fa egyszerűen elvégzi a szeparációt, de a 45°-os rotáció esetében a döntési határ szükségtelenül összetett, nem rendelkezik jó generalizációs tulajdonságokkal.



Instabilitás: variációk az adathalmazban

- Vegyük ki a legszélesebb Versicolor-t (kék) a korábbi modellünkből, és tanítsunk egy fát ugyanazzal az eljárással. Hasonlítsuk össze a modelleket.