# 10. Óra

Adatelemzési platformok, BME, 2018. Március 20., VI. Gyakorlati óra.

# Visszamérési Függvények (folytatás)

## Response chart (illetve cumulative response chart)

- X tengely: algoritmus, cv, pcv, confidencia
  - **primary outcome** confidencia mentén csökkenő sorrendben állítjuk
  - illetve egyenlő elemszámú csoportokra (e.g. decilisekre) bontjuk
- Y tengely: mekkora az egyeseknek az aránya az adott csoportban
  - várhatóan csökkenő tendenciát mutat

Kumulatív verzió: itt azt mutatjuk, hogy az egyesek közül hány százalékot sikerült megtalálnunk.

- Egy jó model esetében még az X tengely vége előtt el kéne érnünk a maximum 1-et
- Illetve ha viszonylag kevés 1-esünk van, akkor már szinte az elején eléri azt

## Lift görbe

- Y: Egyesek aranya az adott csoporton belül / Egyesek aránya az egész adathalmazban
- Kumulatív, és ezért a minimuma (amit a legvégén vesz fel): 1
- A maximuma is kiszámolható, de a paramétektől függ

## Sorbarendezési típusok

Minden eddigi visszamérési görbe a sorbarendezés jóságát vizsgálta

- Az AUC és a GINI egy számba próbálják belesűríteni az egész történetet, ami tipikusabb 'valós' alkalmazásokban és versenyeken. A görbék bonyolultabbak, nehezebben kommunikálhatók

Sorbarendezés viszon akkor hasznos tud lenni, ha nem a pontosság a lényeg, hanem pl az első 20%. Ilyen esetekben használhatóak a chartok is.

[Fakultatív házi: felrajzoltatni a görbéket RM-ben]

## Profitgörbe

Bináris osztályozás, ahol a hiba kétféle lehet (False negative és False positive), különböző találat-típusokhoz különböző 'költségeket' és 'nyereséget' tulajdonítunk.

Profit az egyes esettípusokhoz:

- TP: Nyereség - Költség
- TN: N/A
- FP: -Költség
- FN: -Nyereség

A görbén a konfidencia alapján sorbarendezett esetekhez tartozó profitot kumulálva elkezdjük felírni a profitokat.

- A görbe alakja függ a nyereség és költség egymáshoz képesti arányától is

# Példányalapú osztályozók

## KNN (K-Legközelebbi szomszéd)

- A kapcsolódó model nem a k-means (klaszterezés), hanem k-neighbors (outlier keresés)!

Egy új elem helyét szeretnénk előrejelezni a legközelebbi szomszédok alapján

Módszerek prediktált célváltozó számításra:

1. Többségi döntés
2. Súlyozott módon: a k legközelebbi szomszéd értékeinek 'átlagolása' alapján

1. Emiatt használható regressziós modellre is

Ez egy lusta változó: nincs modell tanítás, rögtön alkalmazás van, ez pedig (a szomszédkeresés miatt) minden egyes új sornál elég nagy számítási költséget generál.

## Problémák és előnyök

- A KNN is hasonló problémákkal küszködik, mint a K-szomszéd modellek.
- Nem túl jól extrapolálható (regressziós esetben látszik a legjobban, gyakorlatilag csak az eredeti adathalmazra lehet alkalmazni)

A KNN előnye azonban (pl a logisztikus regresszióval szemben), hogy az elválasztó 'felület' alakja bármilyen lehet, ezért viszonylag bonyolult dolgokat is ki tud mutatni.

## K meghatározása

- Többségi döntésnél érdemes páratlannak venni
- Minél több, annál aprózottabb lesz
- Nincs ökölszabály erre, érdemes kiszámolni a lehetőségeket

# Döntési fák

- Itt A legfőbb szervező elem a **csomópont**, ami egy bemeneti változó alapján végez egy értékvizsgálatot. (e.g. $BV_i < x_i$)
- A legutolsó elem a fán a **levél**, ahol megtörténik maga a **döntés** (a leggyakoribb érték az adott levélen)

A csomópontokkal a fő cél, hogy a célváltozó szerint homogénebb csoportokat hozzanak létre (e.g. 20%/80% -> 0%-100%). Addig végezzük a vágásokat, amíg 'elég' homogének nem lesznek a csoportok.

## Problémák

- A kiegyensúlyozattlan adathalmaz esetében a döntési fa azt hiszi, hogy jól mér (pl rögtön 0%-100% felosztást csinál).
- A túl mély fa ugyancsak túltanulhatja a modellt, ezért a szinteket is érdemes korlátozni
- Homogenitást ugyancsak úgy tudja kierőltetni, ha elkezd nagyon kis (akár egy) elemszámú ágakat képez: a csoportok nagyságát is érdemes szabályozni.

# CART algoritmus

Classification and regression tree.

$$\phi(S|t) = 2P_L P_R \sum_{j=1}^{CL} |P(j|t_L) - P(j|t_R)|$$

- $t_L$: Bal oldali csomópont
- $t_R$: Jobb oldali csomópont

$$P_L = \frac{t_L sorok}{\sum sor} \quad P_R = \frac{t_R sorok}{\sum sor}$$

$$P(j|t_L) = \frac{sor_j t_L}{\sum sor} \quad P(j|t_R) = \frac{sor_j t_R}{\sum sor}$$

- A $2P_L P_R$ maximális értéke: $2 * 0.5 * 0.5 = 0.5$
- $\sum_{j=1}^{CL} |P(j|t_L) - P(j|t_R)|$ maximális értéke $2$
  - a szalámizást próbálja kiküszöbölni

Együtt az egész értéke 0 és 1 között lehet, és minél nagyobb, annál jobb.

# Változatok

- Entrópia-alapú képletek
- Nem bináris vágásokat végző fák
- Kategória típusú célváltozók vizsgálata
  - Értékcsoportok halmaza alapján
  - Minden értékre egy külön vágás

# Konfidencia

Az egyes levelekhez tartozó konfidenciák alapján csak diszkrét módon tudjuk sorbarendezni az eseteket. A ROC görbe például inkább pontokkal dolgozik.