Paskaita 4

Klasifikavimo ir regresiniai medžiai

Nagrinėdami įvairius algoritmus įsitikinome, kad labai efektyvius (dažnai optimalius) algoritmus pavyksta sukurti ir realizuoti naudojant netiesines duomenų struktūras.

Tarp jų viena populiariausių yra medžio duomenų struktūra.

Šioje paskaitoje klasifikavimo uždavinio sprendimui pasitelkime medžius – sudarysime klasifikavimo medžius (algoritmus

Analogiškai galės būti nagrinėjami ir regresiniai medžiai

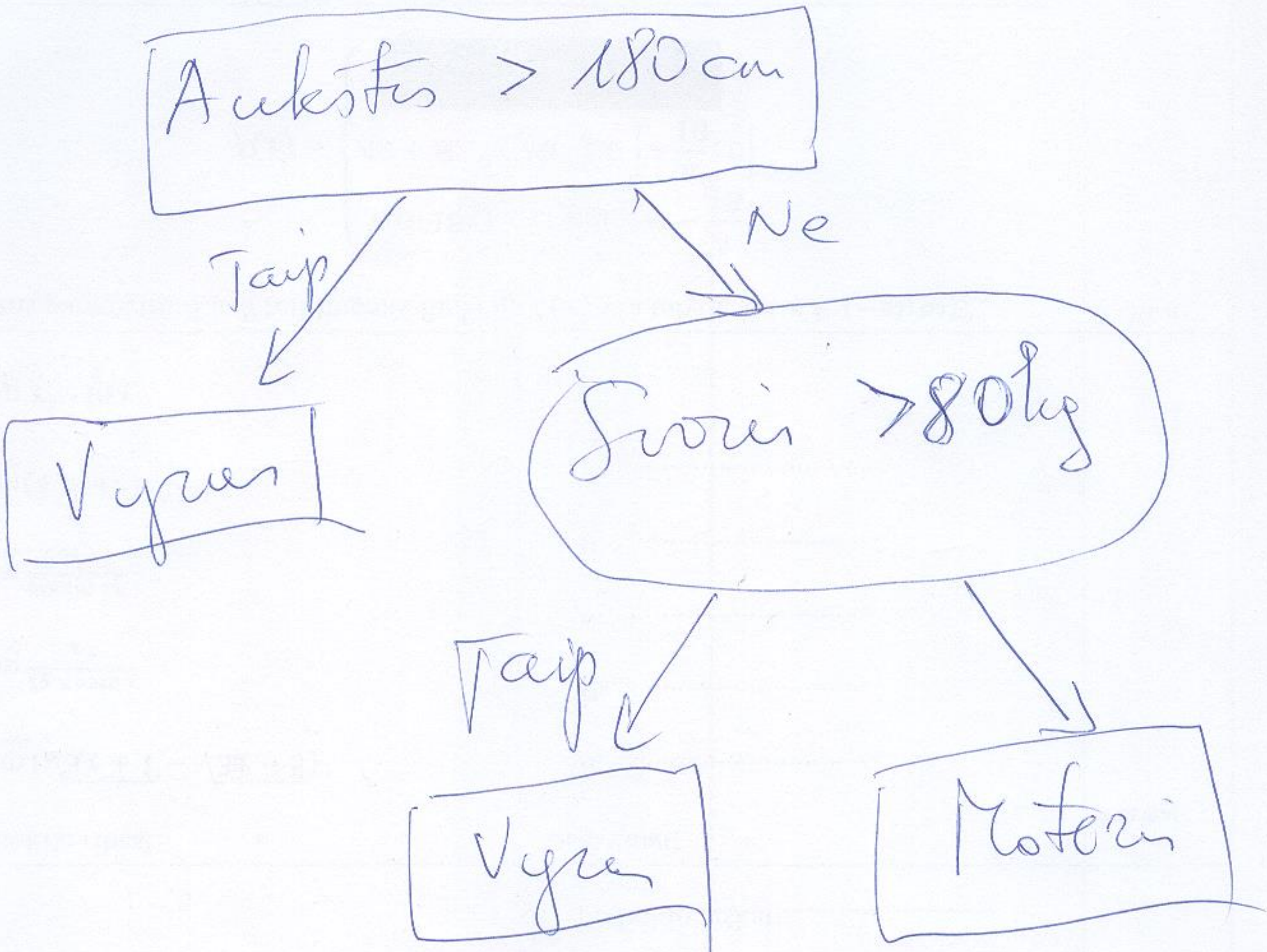
Jie bus naudojami kartu su
mūsų jau išnagrinėtais dažniausiai
klasifikavimo algoritmais. Šios
klasės algoritmus vadiname

CART (Classification And
Regression Trees).

Šio duomenų struktūra padės mums
atlikti/priimti sprendimą, kokiai
klasei priskirsime duomenis, todėl
tokius algoritmus dar vadinsime
sprendimo medžiais (decision trees)

Nagrinėkime pavyzdį, kur
turime dvi klasės Vyrai/Moterys
kejimo kintamųjų parametrai (Ausies,
Svorio)

Klasifikatorius



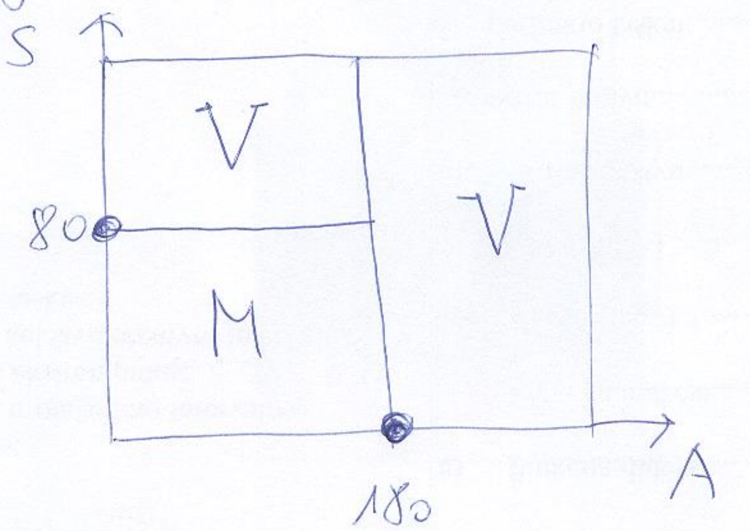
Jeigu turime tokį klasifikatorių (dvejetainį medį) – jį turime išmokti nekonstruoti, tai po galime įvertinti tokius pradinius duomenis (164, 62)

Ats.: ???

Skaidymo žingsniai (apmokymo metu)

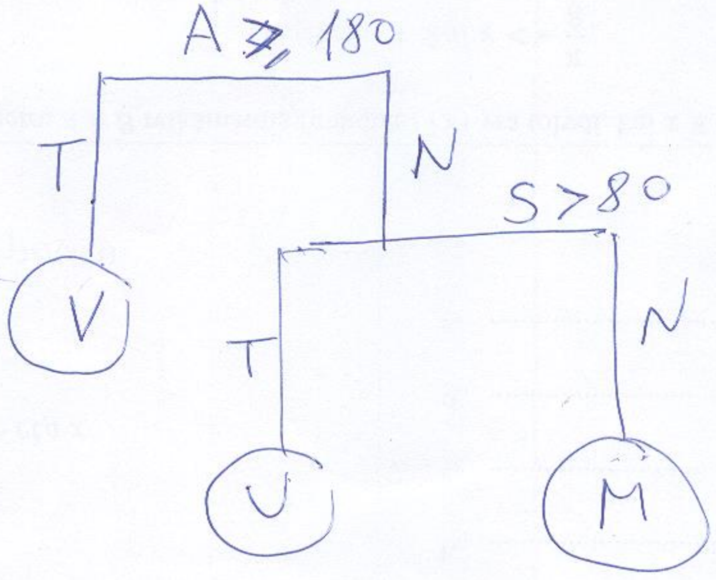
- turime parinkti kitą atliekamą skaidymą
- Kokios yra skaidymo parametrų reikšmės

Ieškome, kas tiksliau aprašytų mokymo duomenų aibę

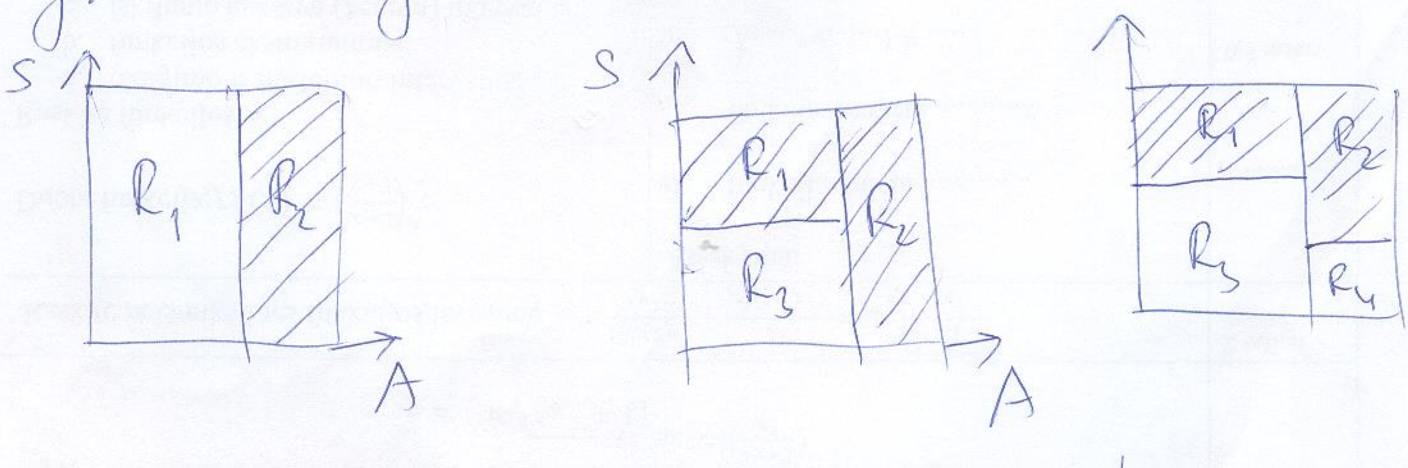


Atvaizduojame prognozių erdvės skaidymą į 3 sritis (2 skaidymo žingsniai)

Regresinius medžius įprasta žymėti (vaizduoti) taip



Geometrinis aiškinimas: Galime nagrinėti dvimatę sritį (aukštis, svoris) ir ją rekursyviai skaidome į sritis R_i



Skaidydamo žingsnyje pasirinktą sritį R skaidome į dvi dalis (sąlygiškai jas vadiname kairiąja (Left) ir dešiniąja (Right)).

Turime K skirtingų klasių (dažnai K = 2 – vyrai ir moterys, taip ir ne).

Pažymėkime

$n_{k,L}$ – duomenų, apibrėžiančių k-klasę ir priklausančių L aibei, skaičių

$n_{k,R}$ – — " — priklausančių R aibei, skaičių

$$n_L = \sum_{k=1}^{K} n_{k,L}, \quad n_R = \sum_{k=1}^{K} n_{k,R},$$

$$p_{k,L} = \frac{n_{k,L}}{n_L}, \quad p_{k,R} = \frac{n_{k,R}}{n_R}.$$

(santykinė dalis k-tosios klasei duomenų).

Apmokymo metu klasifikavimo tikslumą vertiname naudodami Gini funkciją

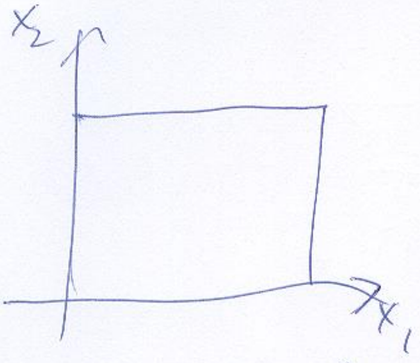
$$G(s_k) = \sum_{j=L,R} \sum_{k=1}^{K} p_{k,j}(1-p_{k,j})$$

Jeigu kažkurioje dalyje j (kairėje ar dešinėje) visos dalelės priklauso tai pačiai klasei, tai tokios daliai $G_j = 0$ - tobulas išskaidymas.

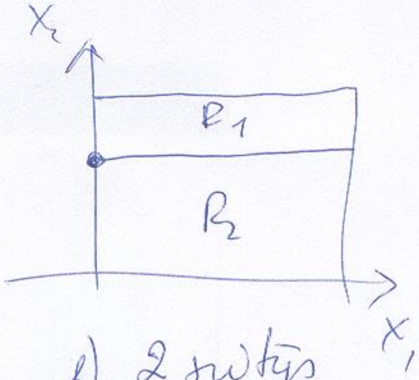
Jeigu, pvz. $K=2$ ir turime mišinį 50:50 procentų, tai tada gauname blogiausią klasifikatorių $G_j = \frac{1}{2}$.

Pasirinkus skaidymo kintamąjį, ieškome geriausios skaidymo vietos (godžioji strategija), nors toks pasirinkimas gali pabloginti galimybę rasti geresnį sprendimą kituose žingsniuose.

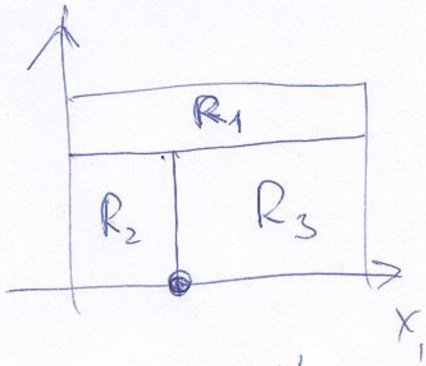
Rekursinis skaidymas 2D atveju



a) prad. sritis



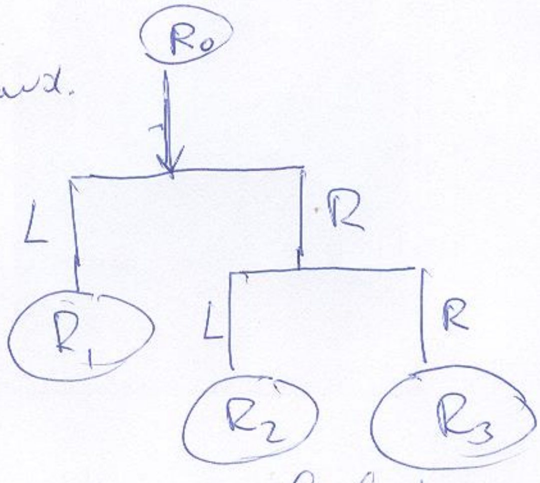
b) 2 sritys po pirmo skaid.



c) trys sritys po antrojo skaid.

- Kiek tokių skaidymo žingsnių reikia atlikti?

?



Skaidymo procesą stabdome, kai viename iš naujų sričių yra mažiau nei N duomenų (pvz. $N \approx 3$ ar 8). Išvengiame persimokymo!

Šakų kapojimo algoritmai

Siekiame, kad gautieji klasifikavimo medžiai būtų kuo paprastesni: jų aukštis ir lapų skaičius būtų kuo mažesnis. Tai padidina ne tik klasifikavimo efektyvumą (skaičiavimų apimtį sumažina), bet algoritmas tampa stabilesnis.

Todėl paskutiniame etape išskiriame iš eilės visus lapus / skaidymus medžio tolimesnėse viršūnėse ir patikriname

- ar pašalinus / tiriamą skaidymą nepablogėja klasifikavimo tikslumas testavimo duomenų aibės atžvilgiu (kaip kada leidžiame šiek tiek tikslumui nuo pablogėti)
- jeigu taip, tada šį skaidymą pašaliname.

Modifikavimas regresinio medžio sudarymui.

Regresijos metodas naudojamas, kai turime prognozuoti funkcijos reikšmę (iš tam tikros srities reikšmių intervalo) naujiems duomenims.

Tada modifikuojame klasifikatoriaus algoritmą:

- kiekvienam suskaidytos srities stačiakampiui priskiriame reikšmę lygią visų tame stačiakampyje esančių duomenų vidurkiui

$$\bar{y}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} y_i ,$$

n_j - duomenų skaičius j-toje srityje R_j

Paklaidą vertiname indeksu:

$$E = \sum_{j=1}^{J} \sum_{i=1}^{n_j} \left(y_i - \bar{y}_j \right)^2 .$$