

Meta  $\tilde{Y}$ 

meta $_X$  vzame podatkovno množico  $S$  sp-spremenljivk in eno ciljno spremenljivko  
dobimo  $\tilde{X} = (\tilde{X}_1, \dots, \tilde{X}_p)$  ... vektor s katerim predložimo podatke  $\tilde{x}$ .

meta $_Y$  preslika množico v eno samo številko  $\tilde{y}$

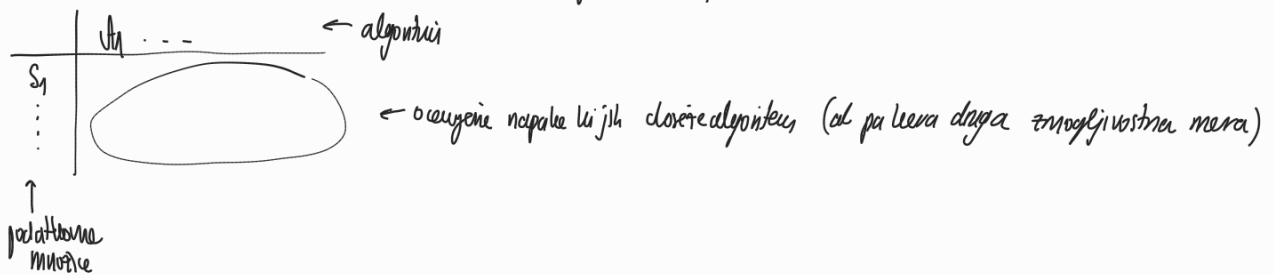
- $\tilde{y} = p(S, A)$   
↑ ocenim napako algoritma ob na podatkovni množici  $S$ .
- $\tilde{Y}$  meta spremenljivka

kako gradimo meta podatkovno množico?

$S_1, \dots, S_n$  podatkovne množice

Podatki za izračun meta $_Y$  52 min.

↳ izračunamo tako da opazujemo zmogljivost algoritmov na podatkovnih množicah



Open ML - repertoar v katerem lahko najdemo podatkovne množice, in podatke lahko ko <sup>različni</sup> algoritmi zmogljivi na tej pod.mn.

Na kaj moramo paziti ko definiramo funkcije meta $_X$ ?

- funkcija mora biti univerzalna in agnatična → vseeno pa je kakšno množico vrže vn.
- funkcija ne sme biti prevelika - drugače se to ne splača delat

$\left[ \argmax p \right]$  — vrne argument pri katerem ima funkcija  $p$  max vrednost.

transfer learning ⇒ vzamemo model in ga doučimo

lahko se za celno sprej. manjši upravičamo pa izračunamo kaj na njej (npr. povprečje, modus, ...)   
  $\uparrow$    
 največja vrednost

[korelacija med  $X_1$  in  $X_2$  0,99; hočemo narediti linearno regresijo. kaj se zgodi?  $\rightarrow$  problematičen invert]   
 visoko korelacija s.s.

$H(X_i) / \log_2 |S| \dots$  normalizirana verjetna entropija

orientirji - označimo ker algoritmiče dobr po tlu

štor - če je štor dobr pot je zelo verjetno da bodo tudi sosednja drevesa dobra   
  $\downarrow$    
 to je da enterot prenehamo misliti pa gledamo kaka je dobr.