

```

1 local R = require
2 local _,the,EGS          = R"lib", R"the", R"egs"
3 local class,OBJ         = _.class, _.OBJ
4
5 local XPLAIN=class("XPLAIN",OBJ)
6 function XPLAIN:new(left,right)
7   local out = {}
8   for n,coln in pairs(left.cols.x) do
9     tmp=coln:bins(right.cols.x[n],BIN)
10    if #tmp > 1 then for _,bin in pairs(tmp) do push(out,bin) end end end end
11
12 local RULE=class("RULE",OBJ)
13 function XPLAIN:asRule(bins)
14   local out={}
15   for _,b in pairs(bins) do out[b.at]=rule[b.at] or {}; push(out[b.at],bin) end
16   return out end
17
18 function XPLAIN:like(rule, klass, h, n) -- h={"true"]=100, "false"]=40) n=100+40
19   fs = {}
20   for at,bins in pairs(rule) do
21     fs[at] = 0
22     for _,bin in pairs(bins) do fs[at] = fs[at] + (bin.ys[klass] or 0) end end
23   local prior = ((h[klass] or 0) + the.K) / (n + the.K * 2)
24   local out = math.log(prior)
25   for _,v in pairs(fs) do out=out+math.log((v+the.M*prior)/(h[klass]+the.M)) end
26   return out end
27
28 local z=1E-32
29 XPLAIN.rules = {}
30 local goal    = XPLAIN.rules
31 function goal.optimize(b,r) return (b<r or (b+r)>.01) and 0 or b^2/(b+r+z) end
32 function goal.monitor( b,r) return (r<b or (b+r)>.01) and 0 or r^2/(b+r+z) end
33 function goal.tabu(    b,r) return 1/(b+r+z) end

```