

Umetna inteligenca

Ocenjevanje atributov



Ocenjevanje atributov

Klasifikacija:

- Informacijski prispevek
- Razmerje informacijskega prispevka
- Mera razdalje
- Gini index
- ReliefF

Regresija:

- Razlika variance



Entropija = mera nečistoče

Količina informacije:

$$I(X_i) = -\log_2 P(X_i)$$

Povprečna pričakovna količina informacije–*entropija*:

$$H(X) = -\sum_i P(X_i) \log_2 P(X_i)$$

H_R – entropija razredov: $H_R = -\sum_k p_k \log p_k$.

H_A – entropija vrednosti danega atributa: $H_A = -\sum_j p_{.j} \log p_{.j}$

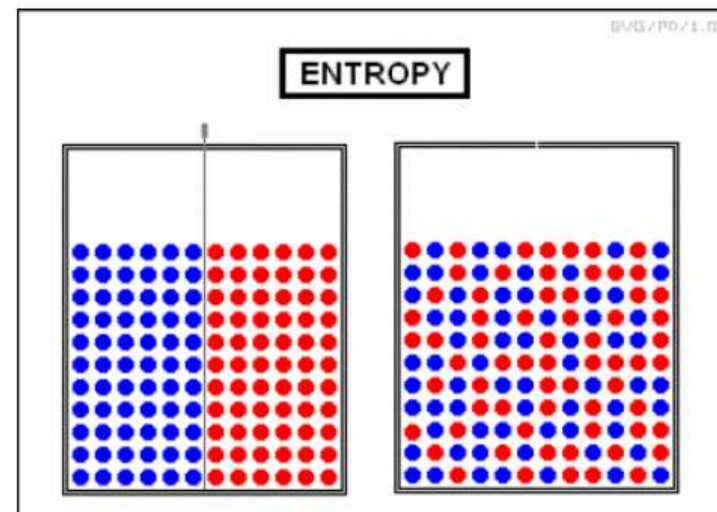
H_{RA} – entropija produkta dogodkov razred–vrednost atributa:

$$H_{RA} = -\sum_k \sum_j p_{kj} \log p_{kj}$$

$H_{R|A}$ – pogojna (pričakovana) entropija razreda pri danem A :

$$H_{R|A} = H_{RA} - H_A = -\sum_j p_{.j} \sum_k \frac{p_{kj}}{p_{.j}} \log \frac{p_{kj}}{p_{.j}}$$

$$H_{R|A} = -\sum_j p_{.j} \sum_k p_{k|j} \log p_{k|j}$$



Informacijski prispevek

$$Gain(A) = H_R + H_A - H_{RA} = H_R - H_{R|A}$$

- Gain(A) je nenegativen
- Maksimalni Gain(A) = H_R
- Slabost: precenjuje večvrednostne attribute



Razmerje informacijskega prispevka

$$GainR(A) = \frac{Gain(A)}{H_A}$$

$$Gain(A_i) \geq \frac{\sum_{j=1}^a Gain(A_j)}{a}$$

- Problematično pri majhnem H_A
- Ad-Hoc
- Najpogosteje uporabljan zaradi C4.5



Mera razdalje dogodkov (atributa od razreda)

$$1 - D(R, A) = \frac{Gain(A)}{H_{RA}}$$

$D(R, A)$ je razdalja:

1. $D(R, A) \geq 0$
2. $D(R, A) = 0 \Leftrightarrow R = A$
3. $D(R, A) = D(A, R)$
4. $D(R, A_1) + D(A_1, A_2) \geq D(R, A_2)$



Gini-index

Apriorni Gini index:

$$Gini_prior = \sum_k \sum_{l \neq k} p_k \cdot p_l = 1 - \sum_k p_k^2$$

Pomembnost: razlika med apriornim in pričakovanim Gini-indexom:

$$Gini(A) = \sum_j p_{\cdot j} \sum_k p_{k|j}^2 - \sum_k p_k^2$$

- Gini_prior je mera nečistoče
- Gini_prior = apriorna verjetnost napačne klasifikacije
- Gini(A) = zmanjšanje verjetnosti napačne klasifikacije
- Gini(A) je nenegativen
- Maksimalni Gini(A) = Gini_prior
- Slabost: precenjuje večvrednostne attribute



ReliefF

- Vse prej opisane mere so kratkovidne
- Ne upoštevajo povezav z drugimi atributi
- Kako bi ocenile attribute A1-A3?

A_1	A_2	A_3	R
0	0	0	0
0	0	1	0
0	1	1	1
0	1	0	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	0	0



Relief Pitcher

Osnovni algoritem RELIEF (Kira & Rendell, 1992)



Relief Pitcher

```
function RELIEF(I: array[1..n] of instance): array[1..a] of real;  
var inst,att : integer; W : array[1..a] of real;  
    M,H : instance; (* najbližji pogrešek, zadetek *)  
begin  
    for att := 1 to a do W[att] := 0.0; end for  
    for inst := 1 to n do  
        primeru I[inst] poišči najbližji pogrešek M in zadetek H;  
        for att := 1 to a do  
            W[att] := W[att] - diff(att,I[inst],H)/n + diff(att,I[inst],M)/n;  
        end for;  
    end for;  
    return(W);  
end;
```

Osnovni algoritem RELIEF (Kira & Rendell, 1992)



Relief Pitcher

$$\text{diff}(A_i, u_j, u_k) = \begin{cases} \frac{|v^{(i,j)} - v^{(i,k)}|}{\text{Max}_i - \text{Min}_i}, & A_i \text{ je zvezni} \\ 0, & v^{(i,j)} = v^{(i,k)} \wedge A_i \text{ je diskretni} \\ 1, & v^{(i,j)} \neq v^{(i,k)} \wedge A_i \text{ je diskretni} \end{cases}$$

Osnovni RELIEF:

- ocenjevanje zveznih in diskretnih atributov
- dvorazredni problemi
- kompleksnost: $O(n^2 \times a) \longrightarrow O(n \times m \times a)$, $m \in [30..200]$

Razširitve v algoritmu ReliefF (Kononenko, 1994)



Relief Pitcher

Neznane vrednosti atributov:

- prvi primer (u_j) nima vrednosti za dani atribut:

$$diff(A_i, u_j, u_k) = 1 - p_{v(i,k)|r(j)}$$

- če oba primera nimata vrednosti za dani primer:

$$diff(A_i, u_j, u_k) = 1 - \sum_{l=1}^{n_i} (p_{V_l|r(j)} \times p_{V_l|r(k)})$$

Šumni podatki: ReliefF poišče k najbližjih zadetkov in k najbližjih pogreškov ter njihove prispevke povpreči.

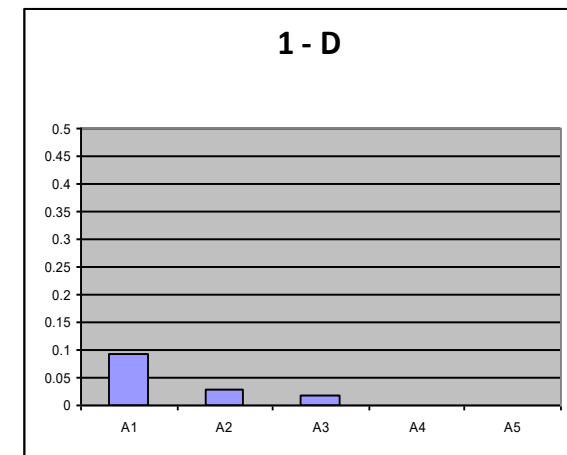
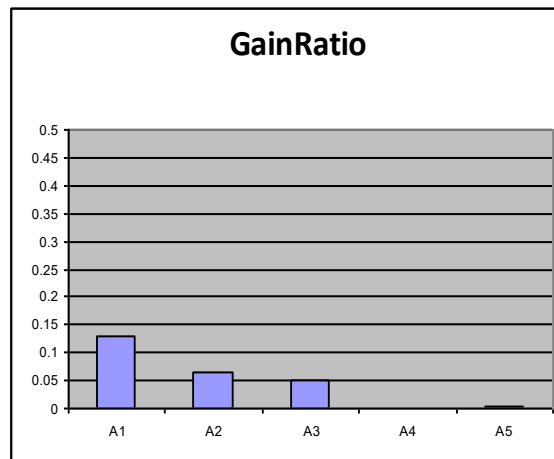
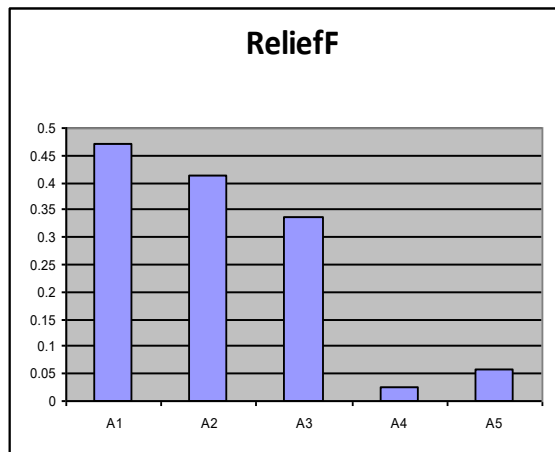
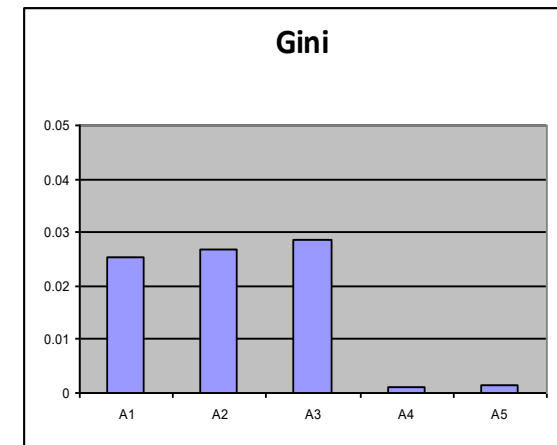
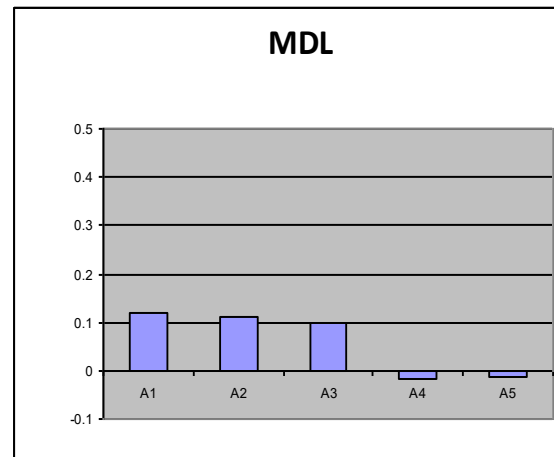
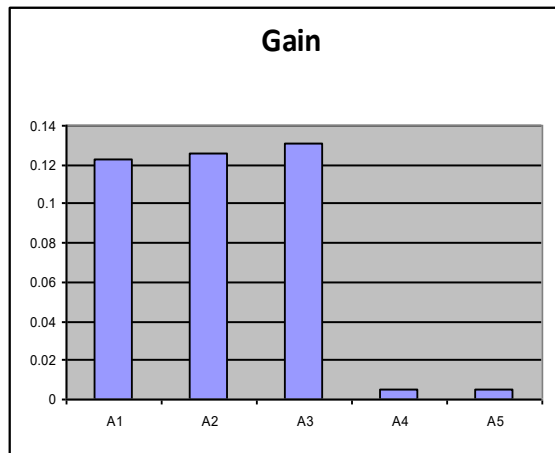
Več razredni problemi: ReliefF poišče k najbližjih pogreškov iz vseh razredov. Prispevki posameznih razredov so dodatno obteženi z apriornimi verjetnostmi razredov

Feature Evaluation



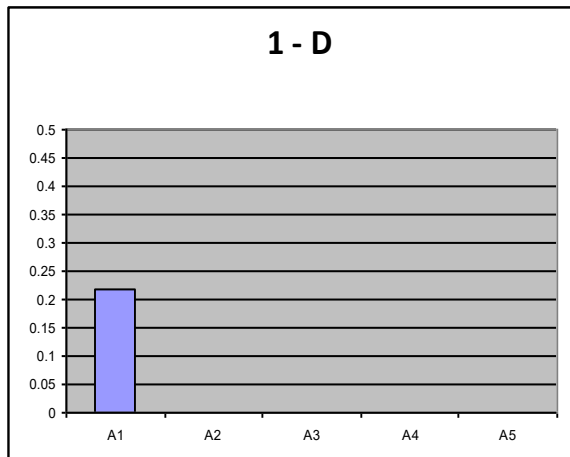
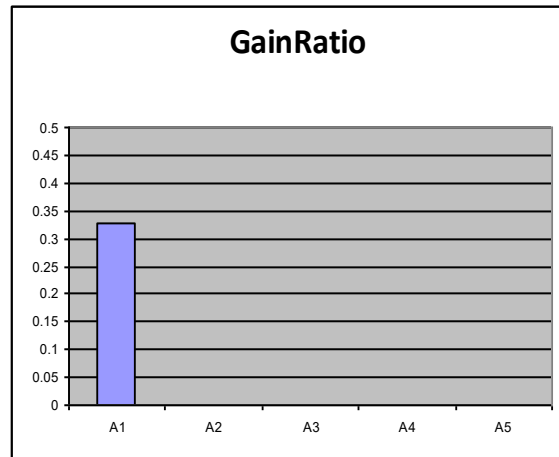
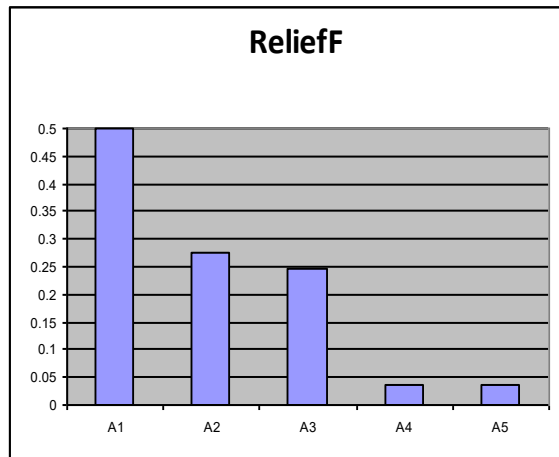
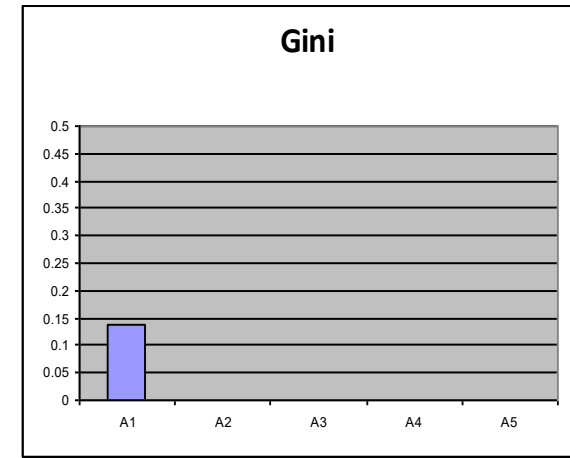
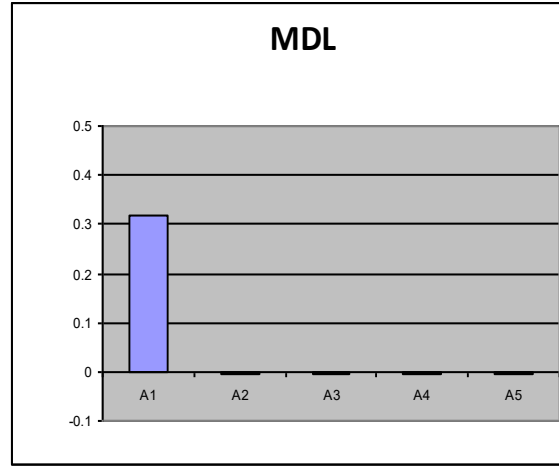
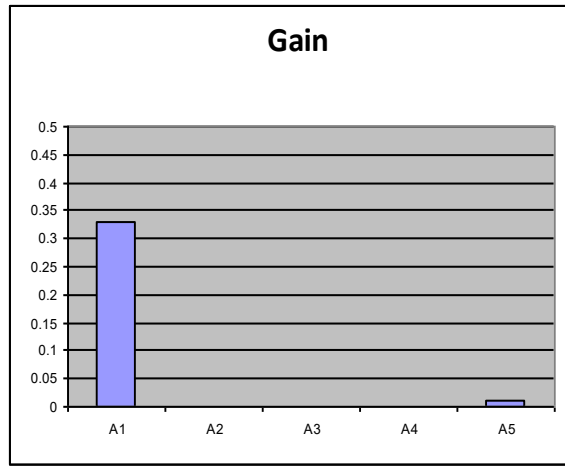
Synthetic Example 1

- 1000 instances, 5 attributes, equiprobable attribute values
- A1: 0,1
- A2: 0,1,2,3
- A3: 0,1,2,3,4,5
- A4: 0,1,2,3,4,5
- A5: 0,1,2,3,4,5
- class value = $(A1 > 0) \text{ OR } (A2 > 1) \text{ OR } (A3 > 2)$



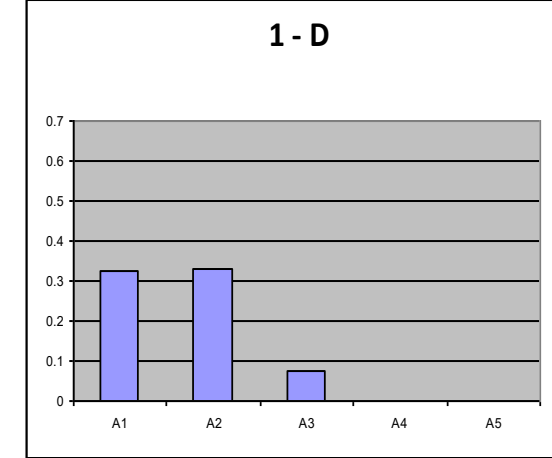
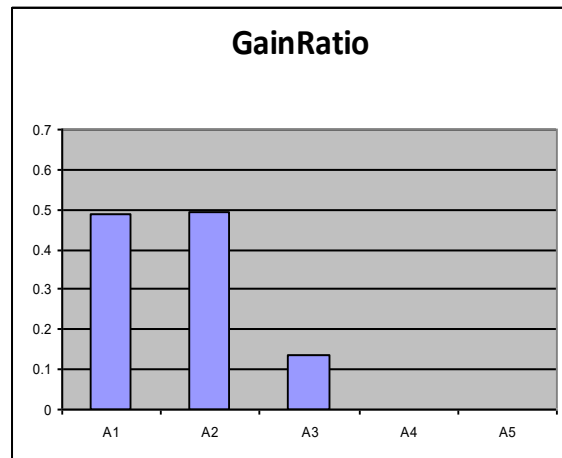
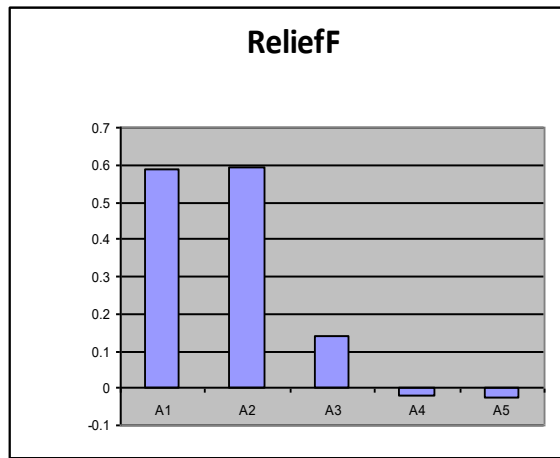
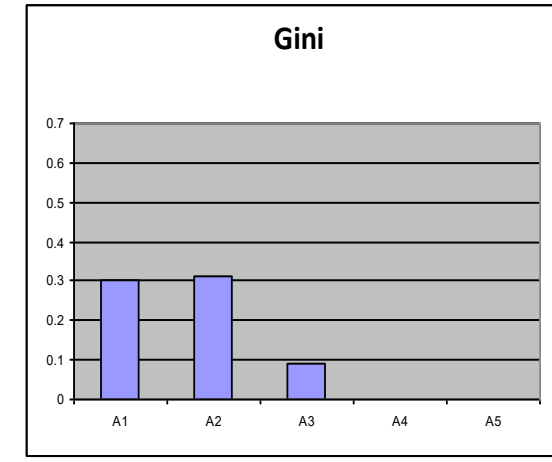
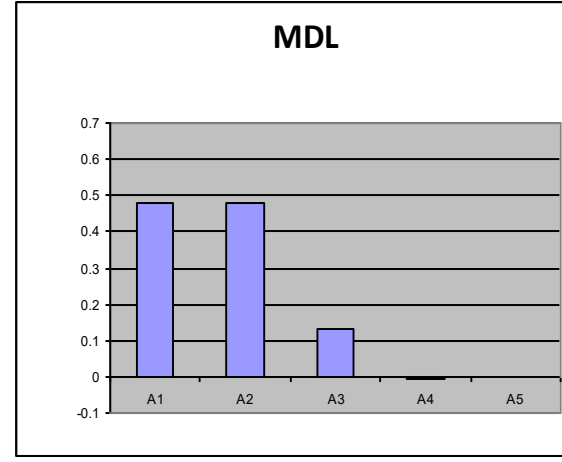
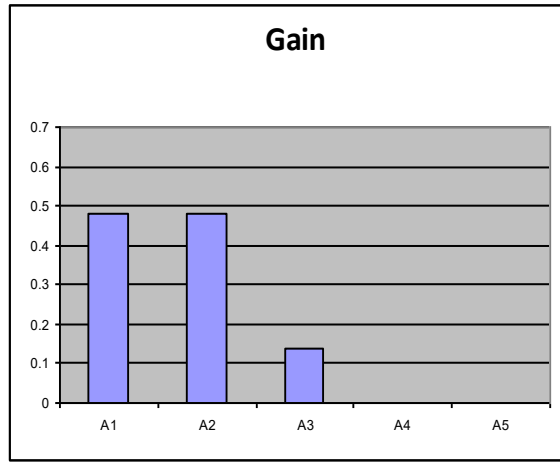
Synthetic Example 2

- 1000 instances, 5 attributes, equiprobable attribute values
- A1: 0,1
- A2: 0,1
- A3: 0,1
- A4: 0,1
- A5: 0,1
- class value = A1 OR (A2 XOR A3)



Synthetic Example 3

- 1000 instances, 5 attributes, 2 equiprobable class values
- A1: 0,1 (is equal to the class value 90% of the time)
- A2: 0,1 (is equal to the class value 90% of the time)
- A3: 0,1 (is equal to the class value 70% of the time)
- A4: 0,1
- A5: 0,1



Regresija: Razlika variance

Pri regresijskih problemih namesto mere nečistoče uporabljamo *varianco zveznega razreda*:

$$s^2 = \frac{1}{n} \sum_{k=1}^n (r^{(k)} - \bar{r})^2$$

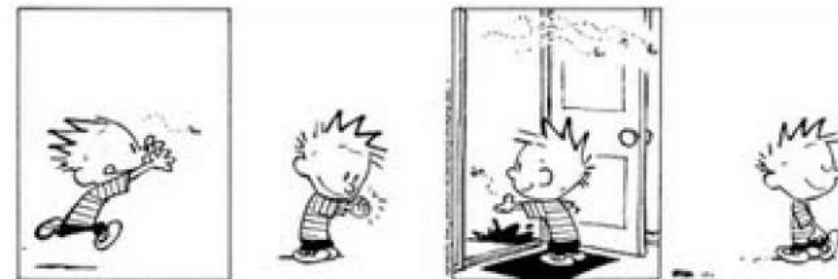
$$\bar{r} = \frac{1}{n} \sum_{k=1}^n r^{(k)}$$

Pomembnost atributa je (nenegativna)
pričakovana razlika variance:

$$ds^2(A_i) = \frac{1}{n} \sum_{k=1}^n (r^{(k)} - \bar{r})^2 - \sum_{j=1}^{n_i} \left(p_{.j} \frac{1}{n_{.j}} \sum_{k=1}^{n_{.j}} (r_j^{(k)} - \bar{r}_j)^2 \right)$$

- Varianca je **mera nečistoče** (zelo podobna Gini_Prior)
- Pričakovana razlika variance je **nenegativna**
- Je **kratkovidna**
- Precenjuje večvrednostne attribute

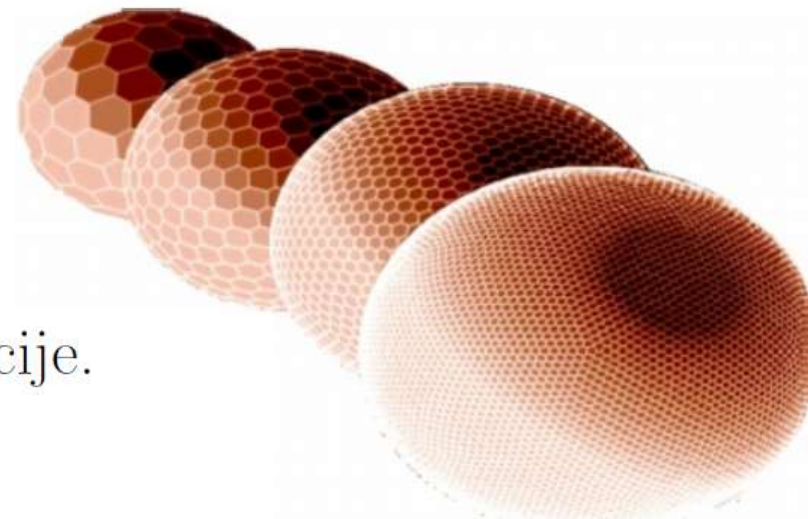
Regression:
"when you fix one bug, you
introduce several newer bugs."



Predprocesiranje podatkov



Diskretizacija zveznih atributov



Vsaka diskretizacija pomeni kvečjemu izgubo informacije.

- optimalno število intervalov
- optimalne meje za vse intervale

Vrste metod za diskretizacijo

Učne primere najprej uredimo po vrednostih zveznega atributa.

- *od spodaj navzgor* (angl. bottom-up)
 n primerov: $O(n^2)$ ali
- *od zgoraj navzdol* (angl. top-down)
 k intervalov: $O(kn)$

Diskretizacija zveznih atributov

- Možne meje: med primeri, ki pripadajo različnim razredom
- Uporaba mer, ki ne precenjujejo večvrednostnih atributov in lahko same ustavijo proces diskretizacije od zgoraj navzdol:
 - Gain Ratio
 - Distance measure
 - ReliefF
- Včasih je koristno uporabiti mehko diskretizacijo
- Binarizacija: izberemo samo eno mejo



Binarizacija atributov

binarna odločitvena drevesa:



- rezultirajoče drevo manjše in s tem optimalnejše:
 - zmanjšanje problema podvajanja (replication problem).
 - preprečuje preveliko razdrobljenost učne množice,
- izognemo se problemu precenjevanja večvrednostnih atributov

Pri binarizaciji **diskretnega atributa**:

- Za vsako vrednost atributa A generiramo en binarni atribut tako, da vse ostale vrednosti združimo v eno vrednost.
- Generiramo toliko binarnih atributov, kolikor je možnih različnih razbitij na dve disjunktni podmnožici:

$$\frac{1}{2} \sum_{k=1}^{n-1} \binom{n}{k} = 2^{n-1} - 1$$

Binarizacija atributov

Dvorazredni problemi

Vrednosti atributa $V_j, j = 1..n$ uredimo po naraščajočih pogojnih verjetnostih prvega razreda:

$$P(r_1|V_1) \leq P(r_1|V_2) \leq \dots \leq P(r_1|V_n)$$

Večrazredni problem:

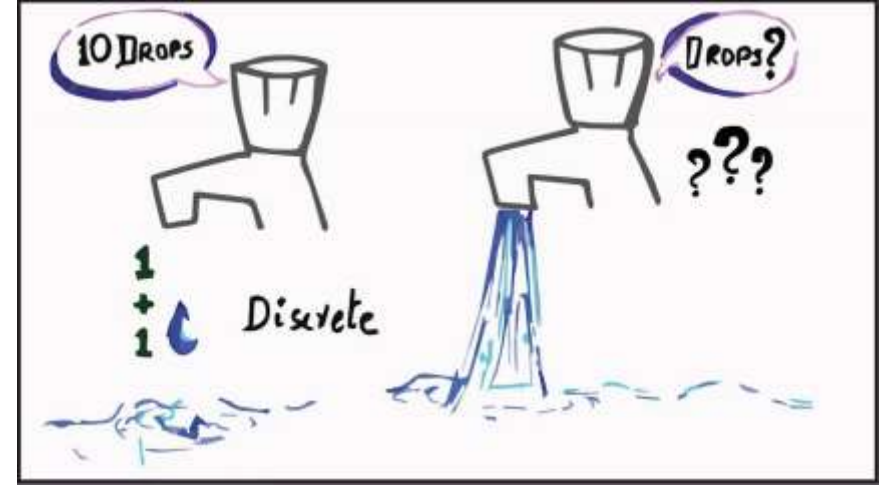
Požrešni algoritem za iskanje suboptimalne binarizacije atributa:

1. Izberi eno vrednost, ki maksimizira kvaliteto atributa, če ga obravnavamo kot binarnega z vsemi ostalimi vrednostmi združenimi v drugo vrednost.
2. Ponavljaj, dokler se kvaliteta atributa ne spreminja več:
 - (a) Če v drugi množici vrednosti obstaja vrednost, ki premaknjena v prvo množico poveča kvaliteto atributa, premakni vrednost, ki maksimizira kvaliteto binarnega atributa.
 - (b) Če v prvi množici vrednosti obstaja vrednost, ki premaknjena v drugo množico poveča kvaliteto atributa, premakni vrednost, ki maksimizira kvaliteto binarnega atributa.



Spreminjanje diskretnih atributov v zvezne

- Vsi atributi morajo biti zvezni: linearna in druge regresije, diskriminantne funkcije in nevronske mreže
- Binarni atribut = poseben primer zveznega atributa.
- Večvrednostne diskretne attribute spremenimo v toliko binarnih atributov, kolikor ima originalni atribut različnih vrednosti.
- Slaba stran: dobimo veliko med seboj odvisnih atributov.



$A = \{\text{sončno, oblačno, deževno, sneži}\}$ →

$A_{\text{sončno}} = \{DA, NE\}$

$A_{\text{oblačno}} = \{DA, NE\}$

$A_{\text{deževno}} = \{DA, NE\}$

$A_{\text{sneži}} = \{DA, NE\}$

Obravnavanje neznanih vrednosti



Če vrednost atributa A_i manjka, ga učna metoda

- ignorira,
 - atributu doda ločeno vrednost (unknown),
 - poskuša nadomestiti manjkajočo vrednost z najbolj verjetno vrednostjo ali
 - uporabi verjetnostno distribucijo po posameznih vrednostih atributa.
- pogojne verjetnosti pri danem razredu primera;
- poseben učni algoritem, ki se nauči preslikave vrednosti ostalih atributov in razreda v atribut brez vrednosti.

Data Preprocessing



"It does data processing, word processing and list processing. Get me some data, some words and some lists."

Dataset: Ecoli

336 instances, default accuracy is approximately 43% (82% in related work)

7 relevant attributes (all are numeric, two can be treated as binary):

1. `mcg`: McGeoch's method for signal sequence recognition
2. `gvh`: von Heijne's method for signal sequence recognition
3. `lip`: von Heijne's Signal Peptidase II consensus sequence score (2 possible values)
4. `chg`: presence of charge on N-terminus of predicted lipoproteins (2 possible values)
5. `aac`: score of analysis of amino acid content (outer membrane & periplasmic proteins)
6. `alm1`: score of the ALOM membrane spanning region prediction program
7. `alm2`: score of ALOM program after excluding putative cleavable signal regions

The class is the protein localization site. Distribution of class values:

<code>cp</code> (cytoplasm)	143
<code>im</code> (inner membrane without signal sequence)	77
<code>pp</code> (periplasm)	52
<code>imU</code> (inner membrane, uncleavable signal sequence)	35
<code>om</code> (outer membrane)	20
<code>omL</code> (outer membrane lipoprotein)	5
<code>imL</code> (inner membrane lipoprotein)	2
<code>imS</code> (inner membrane, cleavable signal sequence)	2

Some Models Require Discrete Data

We want to use a naive Bayes classifier. We use leave-one-out cross-validation to evaluate its performance:

mean accuracy = 71.4% is not as good as we would expect,... why?

NB treats each distinct numeric value as a discrete attribute value.

	A1	A2	A3	A4	A5	A6	A7	class
#58	0.40	0.35	0.48	0.5	0.45	0.33	0.42	cp
#272	0.65	0.51	0.48	0.5	0.66	0.54	0.33	om
#130	0.37	0.44	0.48	0.5	0.42	0.39	0.47	cp
#111	0.32	0.33	0.48	0.5	0.60	0.06	0.20	cp
#201	0.58	0.55	0.48	0.5	0.57	0.70	0.74	im
#202	0.36	0.47	0.48	0.5	0.51	0.69	0.72	im
#43	0.40	0.50	0.48	0.5	0.45	0.39	0.47	cp
#98	0.57	0.54	0.48	0.5	0.37	0.28	0.33	cp
#191	0.33	0.37	0.48	0.5	0.46	0.65	0.69	im
#208	0.11	0.50	0.48	0.5	0.58	0.72	0.68	im

Class Attribute	cp (0.42)	im (0.23)	imL (0.01)	imS (0.01)	imU (0.1)	om (0.06)	omL (0.02)	pp (0.15)
=====								
A1								
0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.04	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.06	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.07	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.11	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.12	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0
0.16	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.17	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.18	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.2	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
			.	.	.			
			.	.	.			
0.81	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0
0.83	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0
0.84	0.0	1.0	0.0	0.0	2.0	0.0	0.0	0.0
0.85	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.86	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0
0.87	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.88	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.89	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
[total]	143.0	77.0	52.0	35.0	20.0	5.0	2.0	2.0

We get too many coefficients and very sparse data.



Discretization

A better solution: Discretization

	A1	A2	A3	A4	A5	A6	A7	class
#58	0.40	0.35	0.48	0.5	0.45	0.33	0.42	cp
#272	0.65	0.51	0.48	0.5	0.66	0.54	0.33	om
#130	0.37	0.44	0.48	0.5	0.42	0.39	0.47	cp
#111	0.32	0.33	0.48	0.5	0.60	0.06	0.20	cp
#201	0.58	0.55	0.48	0.5	0.57	0.70	0.74	im
#202	0.36	0.47	0.48	0.5	0.51	0.69	0.72	im
#43	0.40	0.50	0.48	0.5	0.45	0.39	0.47	cp
#98	0.57	0.54	0.48	0.5	0.37	0.28	0.33	cp
#191	0.33	0.37	0.48	0.5	0.46	0.65	0.69	im
#208	0.11	0.50	0.48	0.5	0.58	0.72	0.68	im

Intervals of equal width.

	A1	A2	A3	A4	A5	A6	A7	class
#58	5	3	1	1	8	3	5	cp
#272	8	6	1	1	12	5	4	om
#130	5	5	1	1	8	4	5	cp
#111	4	3	1	1	11	1	3	cp
#201	7	6	1	1	10	7	8	im
#202	5	5	1	1	9	7	8	im
#43	5	5	1	1	8	4	5	cp
#98	7	6	1	1	7	3	4	cp
#191	4	4	1	1	8	6	7	im
#208	2	5	1	1	10	7	7	im

accuracy improves from 71.4% to 81.8%

More Discretization

Alternative discretization methods:

	A1	A2	A3	A4	A5	A6	A7	class
#58	0.40	0.35	0.48	0.5	0.45	0.33	0.42	cp
#272	0.65	0.51	0.48	0.5	0.66	0.54	0.33	om
#130	0.37	0.44	0.48	0.5	0.42	0.39	0.47	cp
#111	0.32	0.33	0.48	0.5	0.60	0.06	0.20	cp
#201	0.58	0.55	0.48	0.5	0.57	0.70	0.74	im
#202	0.36	0.47	0.48	0.5	0.51	0.69	0.72	im
#43	0.40	0.50	0.48	0.5	0.45	0.39	0.47	cp
#98	0.57	0.54	0.48	0.5	0.37	0.28	0.33	cp
#191	0.33	0.37	0.48	0.5	0.46	0.65	0.69	im
#208	0.11	0.50	0.48	0.5	0.58	0.72	0.68	im

Equal width (Scott's formula).

	A1	A2	A3	A4	A5	A6	A7	class
#58	5	3	1	1	8	3	5	cp
#272	8	6	1	1	12	5	4	om
#130	5	5	1	1	8	4	5	cp
#111	4	3	1	1	11	1	3	cp
#201	7	6	1	1	10	7	8	im
#202	5	5	1	1	9	7	8	im
#43	5	5	1	1	8	4	5	cp
#98	7	6	1	1	7	3	4	cp
#191	4	4	1	1	8	6	7	im
#208	2	5	1	1	10	7	7	im

accuracy improves from 71.4% to 81.8%

Binarization.

	A1	A1.1	A2	A3	A4	A5	A5.1	A6	A6.1	A7	class
58	0	0	0	0	1	0	0	0	0	0	cp
272	1	0	0	0	1	1	0	1	0	0	om
130	0	0	0	0	1	0	0	1	0	0	cp
111	0	0	0	0	1	1	0	0	0	0	cp
201	1	0	0	0	1	1	0	1	1	1	im
202	0	0	0	0	1	0	0	1	1	1	im
43	0	0	0	0	1	0	0	1	0	0	cp
98	1	0	0	0	1	0	0	0	0	0	cp
191	0	0	0	0	1	0	0	1	1	1	im
208	0	0	0	0	1	1	0	1	1	1	im

accuracy improves from 71.4% to 85.5%

Min. entropy (MDL).

	A1	A2	A3	A4	A5	A6	A7	class
58	1	1	1	1	1	1	1	cp
272	2	1	1	1	2	2	1	om
130	1	1	1	1	1	2	1	cp
111	1	1	1	1	2	1	1	cp
201	2	1	1	1	2	3	2	im
202	1	1	1	1	1	3	2	im
43	1	1	1	1	1	2	1	cp
98	2	1	1	1	1	1	1	cp
191	1	1	1	1	1	3	2	im
208	1	1	1	1	2	3	2	im

accuracy improves from 71.4% to 85.4%

Let's Throw Away Some Information

We replace 300 features values with unknown values or NA's (at random).


	A1	A2	A3	A4	A5	A6	A7	class			A1	A2	A3	A4	A5	A6	A7	class	
#58	5	3	1	1	8	3	5	cp			class	NA	3	1	1	8	3	NA	cp
#272	8	6	1	1	12	5	4	om			#58	NA	3	1	1	8	3	NA	cp
#130	5	5	1	1	8	4	5	cp			#272	8	6	1	1	12	5	4	om
#111	4	3	1	1	11	1	3	cp			#130	5	NA	1	1	8	NA	5	cp
#201	7	6	1	1	10	7	8	im			#111	4	3	1	1	NA	1	NA	cp
#202	5	5	1	1	9	7	8	im			#201	7	6	1	1	10	7	8	im
#43	5	5	1	1	8	4	5	cp			#202	5	5	1	1	9	NA	NA	im
#98	7	6	1	1	7	3	4	cp			#43	5	5	1	1	8	4	5	cp
#191	4	4	1	1	8	6	7	im			#98	7	6	1	1	7	3	4	cp
#208	2	5	1	1	10	7	7	im			#191	4	4	NA	1	8	6	7	im
											#208	2	5	1	1	10	7	NA	im

accuracy decreases from 81.8% to 77.1%

By omitting instances with at least one NA from the training data,
we lose about 200 instances.

Treating Missing Values (1)

	A1	A2	A3	A4	A5	A6	A7	class
#58	NA	3	1	1	8	3	NA	cp
#272	8	6	1	1	12	5	4	om
#130	5	NA	1	1	8	NA	5	cp
#111	4	3	1	1	NA	1	NA	cp
#201	7	6	1	1	10	7	8	im
#202	5	5	1	1	9	NA	NA	im
#43	5	5	1	1	8	4	5	cp
#98	7	6	1	1	7	3	4	cp
#191	4	4	NA	1	8	6	7	im
#208	2	5	1	1	10	7	NA	im



Treating NA's as a special value.

	A1	A2	A3	A4	A5	A6	A7	class
#58	NA	3	1	1	8	3	NA	cp
#272	8	6	1	1	12	5	4	om
#130	5	NA	1	1	8	NA	5	cp
#111	4	3	1	1	NA	1	NA	cp
#201	7	6	1	1	10	7	8	im
#202	5	5	1	1	9	NA	NA	im
#43	5	5	1	1	8	4	5	cp
#98	7	6	1	1	7	3	4	cp
#191	4	4	NA	1	8	6	7	im
#208	2	5	1	1	10	7	NA	im

accuracy further decreases from 77.1% to 61.6%

Replacing with most frequent value

	A1	A2	A3	A4	A5	A6	A7	class
#58	8	3	1	1	8	3	4	cp
#272	8	6	1	1	12	5	4	om
#130	5	5	1	1	8	3	5	cp
#111	4	3	1	1	9	1	4	cp
#201	7	6	1	1	10	7	8	im
#202	5	5	1	1	9	3	4	im
#43	5	5	1	1	8	4	5	cp
#98	7	6	1	1	7	3	4	cp
#191	4	4	1	1	8	6	7	im
#208	2	5	1	1	10	7	4	im

accuracy increases from 77.1% to 80.7%

Treating Missing Values (2)

A test instance contains a missing value:

	A1	A2	A3	A4	A5	A6	A7	class
#98	7	NA	1	1	7	3	4	?

Replace with most common value:

	A1	A2	A3	A4	A5	A6	A7	class
#98	7	5	1	1	7	3	4	?

OR

Predicted class: "cp" ($p_{cp} = 0.96$)

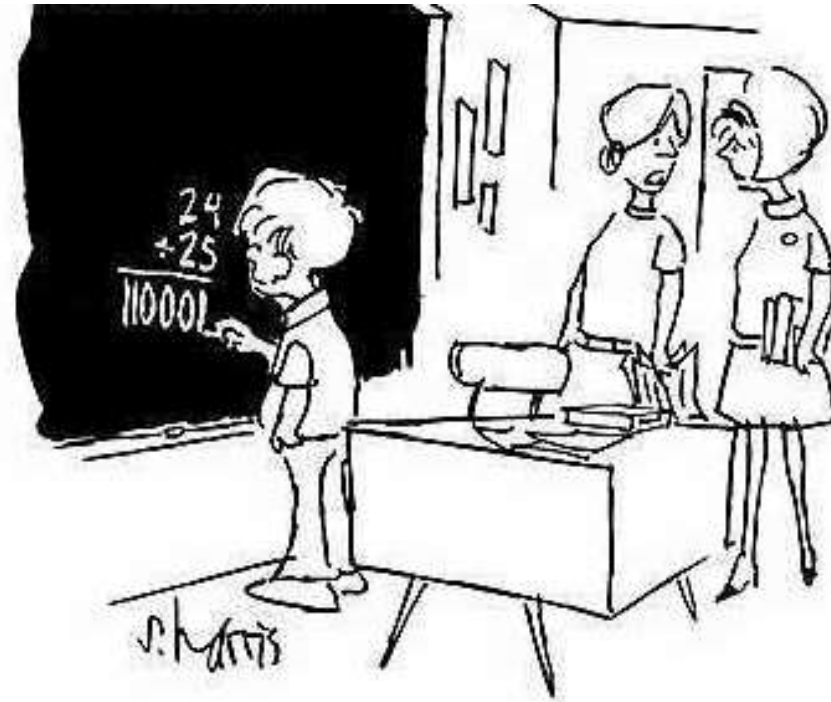
Use weighted sum of predictions across all possible values:

	A1	A2	A3	A4	A5	A6	A7	class	predicted class (prob.)	P(A2 = x)
#98	7	1	1	1	7	3	4	?	"cp" ($p_{cp} = 0.83$)	0.003
#98	7	2	1	1	7	3	4	?	"cp" ($p_{cp} = 0.97$)	0.046
#98	7	3	1	1	7	3	4	?	"cp" ($p_{cp} = 0.98$)	0.102
#98	7	4	1	1	7	3	4	?	"cp" ($p_{cp} = 0.98$)	0.215
#98	7	5	1	1	7	3	4	?	"cp" ($p_{cp} = 0.96$)	0.222
#98	7	6	1	1	7	3	4	?	"cp" ($p_{cp} = 0.90$)	0.182
#98	7	7	1	1	7	3	4	?	"pp" ($p_{cp} = 0.20$)	0.062
#98	7	8	1	1	7	3	4	?	"pp" ($p_{cp} = 0.20$)	0.061
#98	7	9	1	1	7	3	4	?	"pp" ($p_{cp} = 0.39$)	0.043
#98	7	10	1	1	7	3	4	?	"pp" ($p_{cp} = 0.15$)	0.055
#98	7	11	1	1	7	3	4	?	"cp" ($p_{cp} = 0.66$)	0.006
#98	7	12	1	1	7	3	4	?	"cp" ($p_{cp} = 0.57$)	0.003
Predicted class: 0.66 times "cc" OR $p'_{cp} = 0.79$										

More complicated when several values are missing.

The same approach can easily be used to treat missing values on the training set (if the classifier supports weighted examples).

Feature Binarization



"It was bound to happen—they're beginning to think like binary computers."

Regression, Non-Continuous Features

Example dataset

(sensory data, 576 instances, 11 features, continuous class):

```
Occasion {1, 2}
Judges {1, 2, 3, 4, 5, 6}
Interval {1, 2, 3}
Sittings {1, 2, 3, 4}
Position {1, 2, 3, 4}
Squares {1, 2}
Rows {1, 2, 3}
Columns {1, 2, 3, 4}
Halfplot {1, 2}
Trellis {1, 2, 3, 4}
Method {1, 2}
CLASS real
```

One of the most basic approaches is using linear regression.
However, we need numeric attributes, not nominal.

Simple solution = straightforward transformation of feature values to discrete numeric values. Results (using 10-fold CV):

```
Correlation coefficient    0.1229
Mean absolute error       0.6583
Root mean squared error   0.8189
```

Linear Regression Model

score =

-0.0836 * Judges +

-0.0768 * Rows +

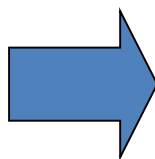
0.0528 * Trellis +

15.3891

Regression, Binary Features

Alternative solution = transform each value of a nominal feature to a binary feature:

Occasion {1, 2}
Judges {1, 2, 3, 4, 5, 6}
Interval {1, 2, 3}
Sittings {1, 2, 3, 4}
Position {1, 2, 3, 4}
Squares {1, 2}
Rows {1, 2, 3}
Columns {1, 2, 3, 4}
Halfplot {1, 2}
Trellis {1, 2, 3, 4}
Method {1, 2}
score numeric



Occasion numeric
Judges=1 numeric
Judges=2 numeric
Judges=3 numeric
Judges=4 numeric
Judges=5 numeric
Judges=6 numeric
Interval=1 numeric
Interval=2 numeric
Interval=3 numeric
Sittings=1 numeric
Sittings=2 numeric
Sittings=3 numeric
Sittings=4 numeric
Position=1 numeric
Position=2 numeric
Position=3 numeric
Position=4 numeric

Squares numeric
Rows=1 numeric
Rows=2 numeric
Rows=3 numeric
Columns=1 numeric
Columns=2 numeric
Columns=3 numeric
Columns=4 numeric
Halfplot numeric
Trellis=1 numeric
Trellis=2 numeric
Trellis=3 numeric
Trellis=4 numeric
Method numeric
score numeric

Results:

Linear Regression Model

score =

0.2656 * Judges=2 +
0.1719 * Judges=3 +
-0.224 * Judges=4 +
-0.2969 * Judges=6 +
0.1068 * Interval=2 +
-0.1644 * Position=2 +
0.4167 * Rows=2 +
-0.1536 * Rows=3 +
-0.2778 * Trellis=2 +
0.1875 * Trellis=3 +
15.0289

A more complex model.

Correlation coefficient	0.1229
Mean absolute error	0.6583
Root mean squared error	0.8189



Better results.

Correlation coefficient	0.3822
Mean absolute error	0.6088
Root mean squared error	0.7618

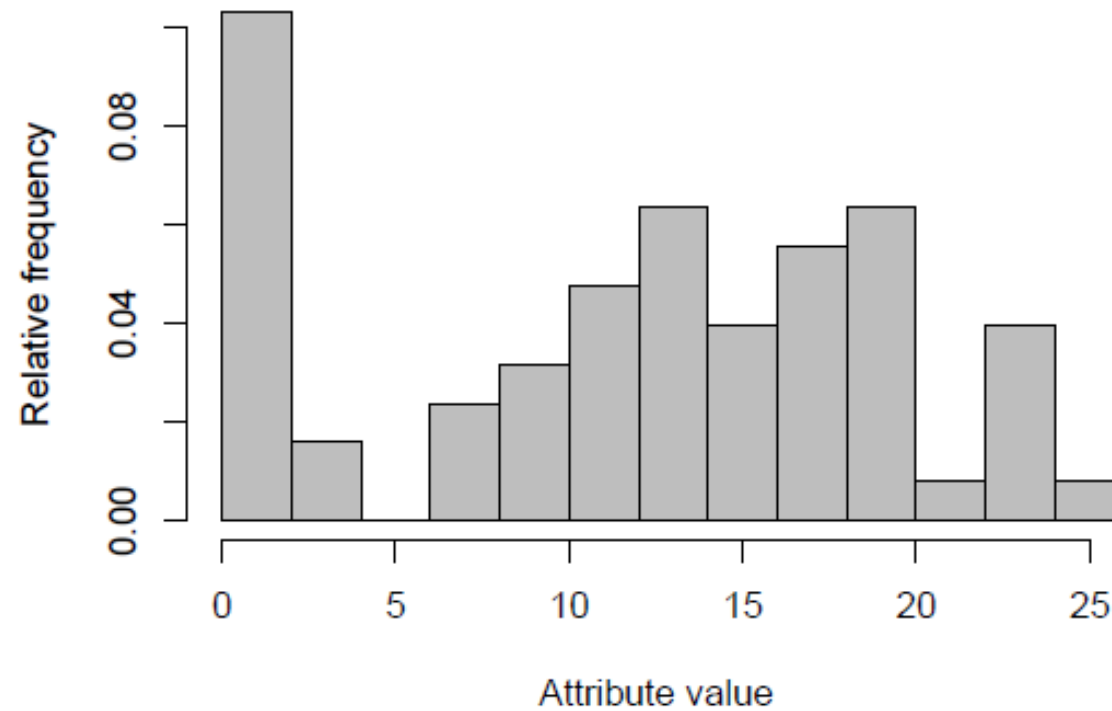
VISUALIZATION OF DATA

- ✗ Humans are good in visual pattern recognition
- ✗ Discovery of regularities in data
- ✗ Examples:
 - + Rules for separating different classes
 - + Clusters
 - + Outliers
 - + Interactions among attributes
 - + Structures and trends in data
 - + ...

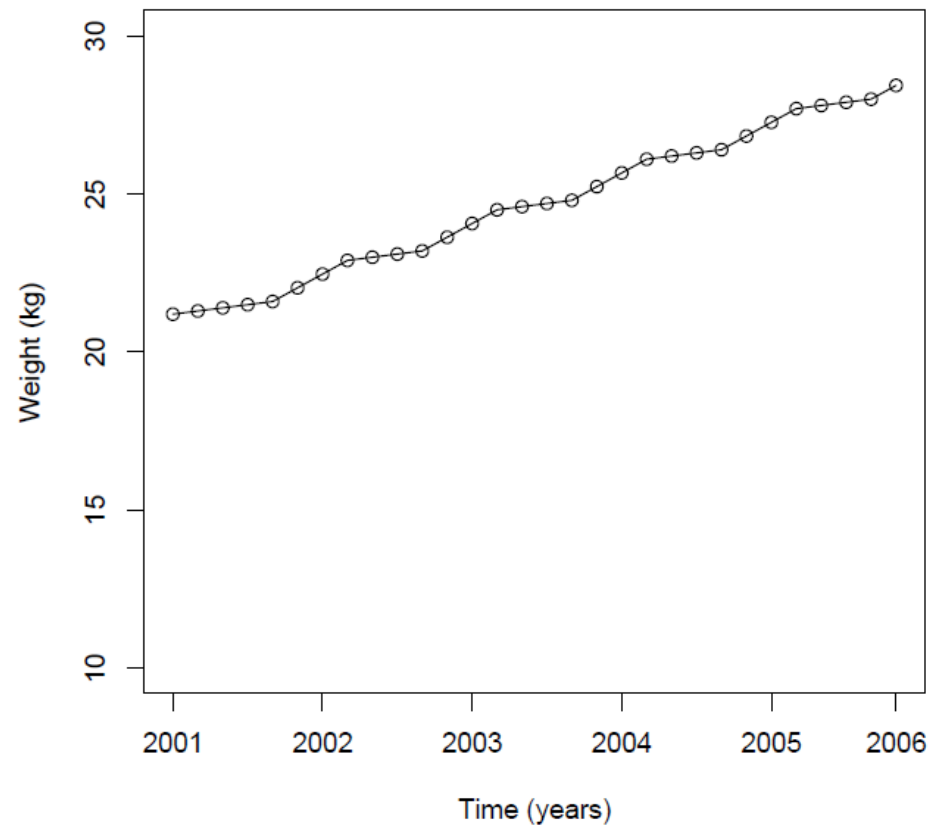
PROBLEMS WITH VISUALIZATION

- ✗ Large number of possible visualizations
- ✗ Example: *scatterplot (visualization of 2 attributes)*
 - ✗ 100atts $\rightarrow (100 * 99) / 2 = 4950$ different diagrams
 - + Data from bioinformatics:
 - 5.000atts $\rightarrow >12$ milion different diagrams
- ✗ Manual search for interesting visualizations is impossible.
- ✗ Computers can automatically search for interesting visualizations.

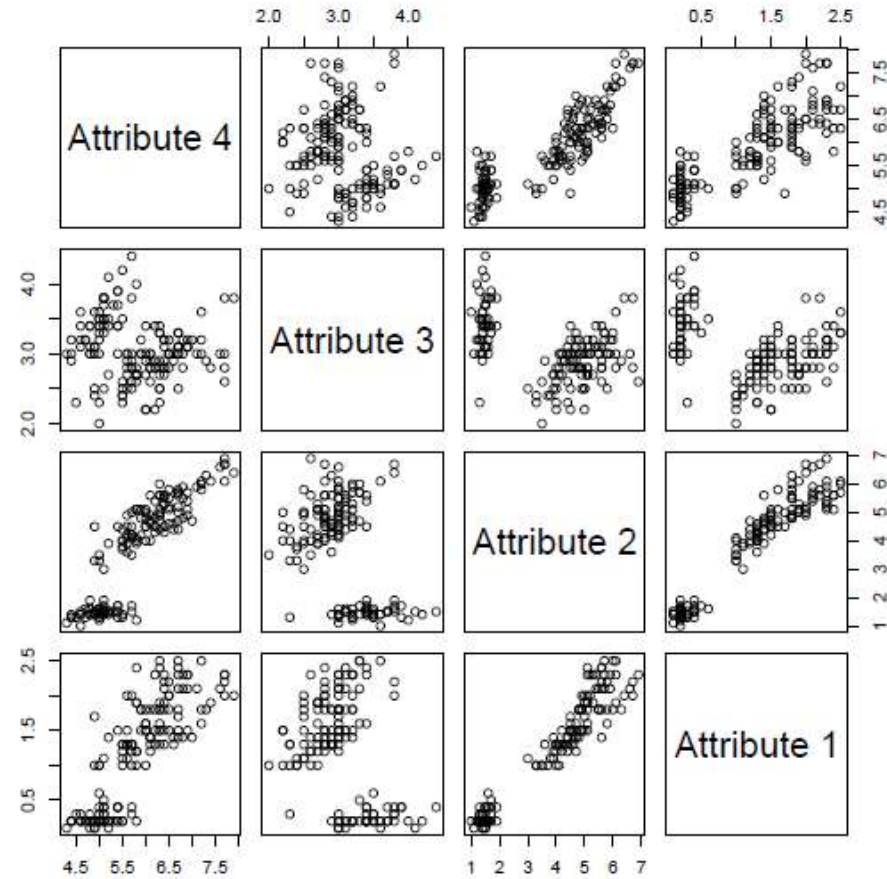
VISUALIZATION METHODS: SINGLE ATT



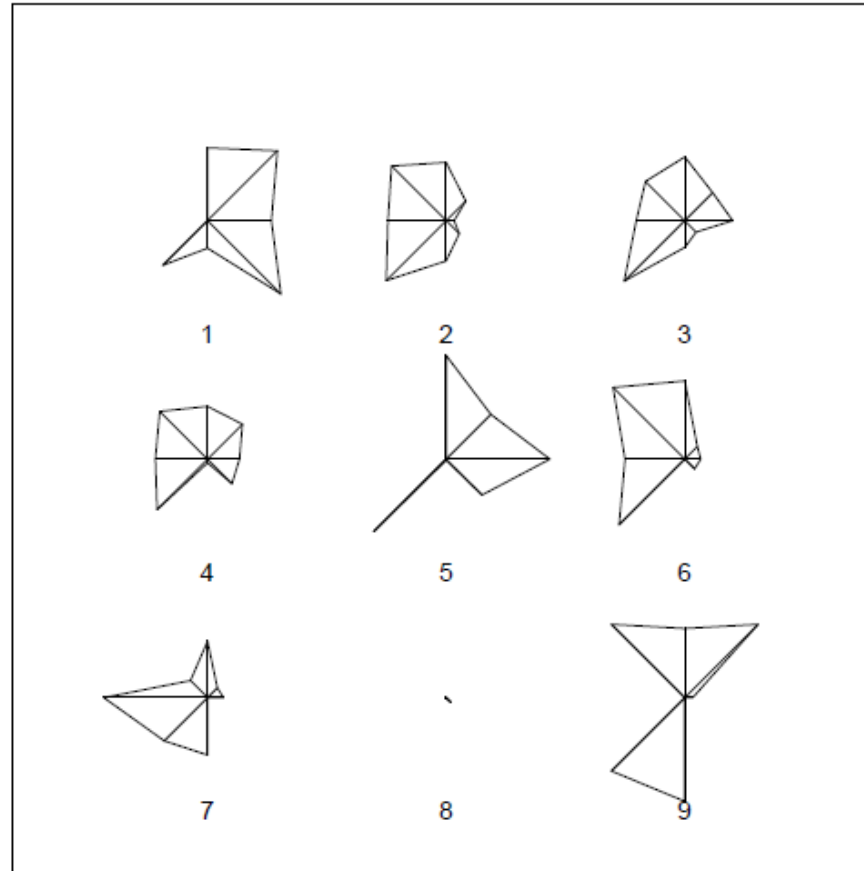
VISUALIZATION: SINGLE ATT VS TIME



VISUALIZATION METHODS: SCATTERPLOT MATRIX – PAIRWISE DEPENDENCIES

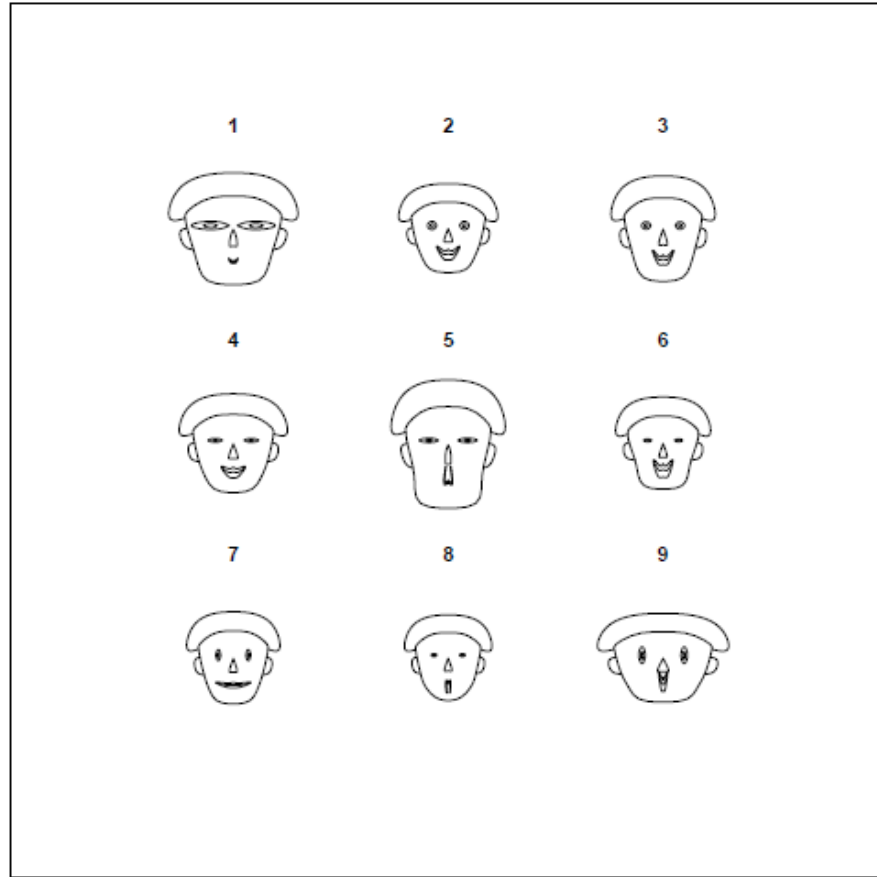


VISUALIZATION METHODS: STAR GLYPH PLOT

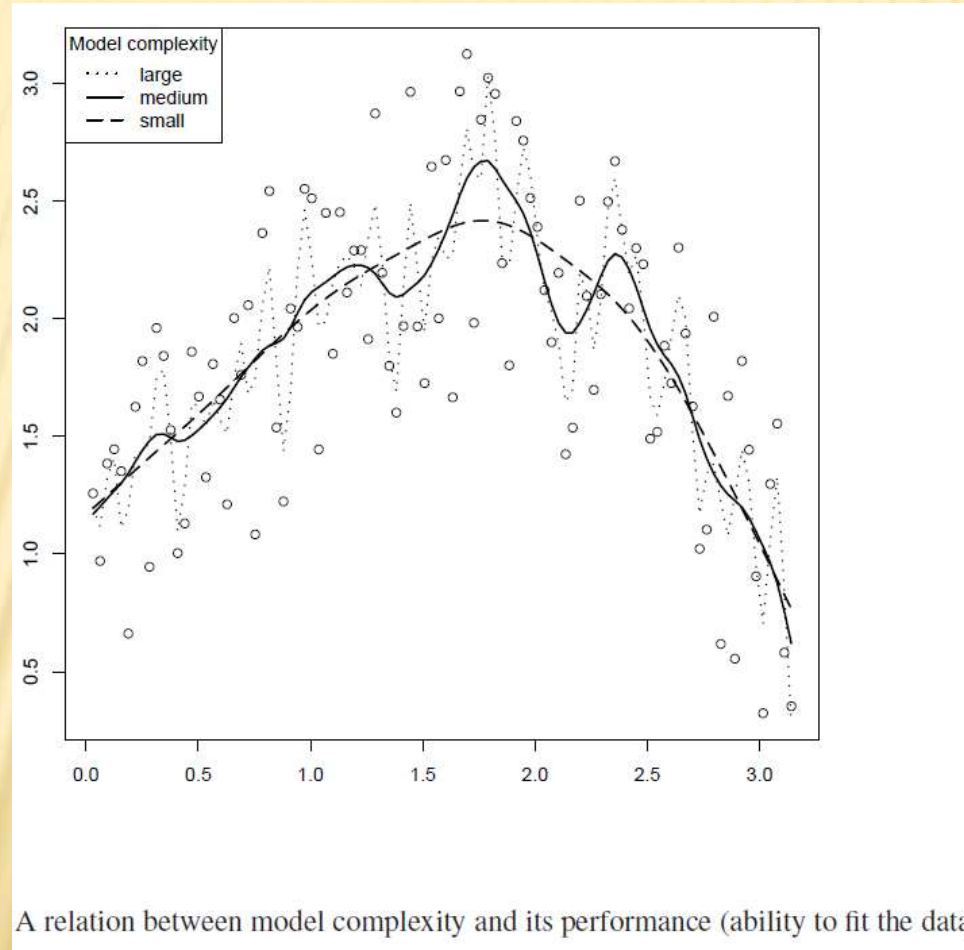


VISUALIZATION METHODS: FACE PLOT

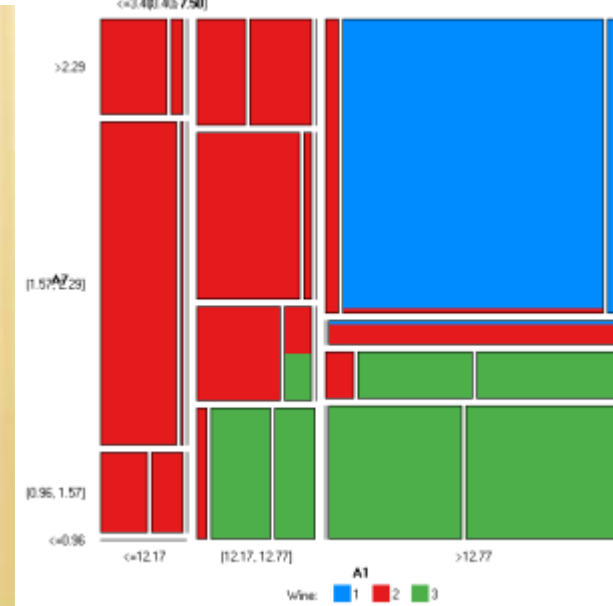
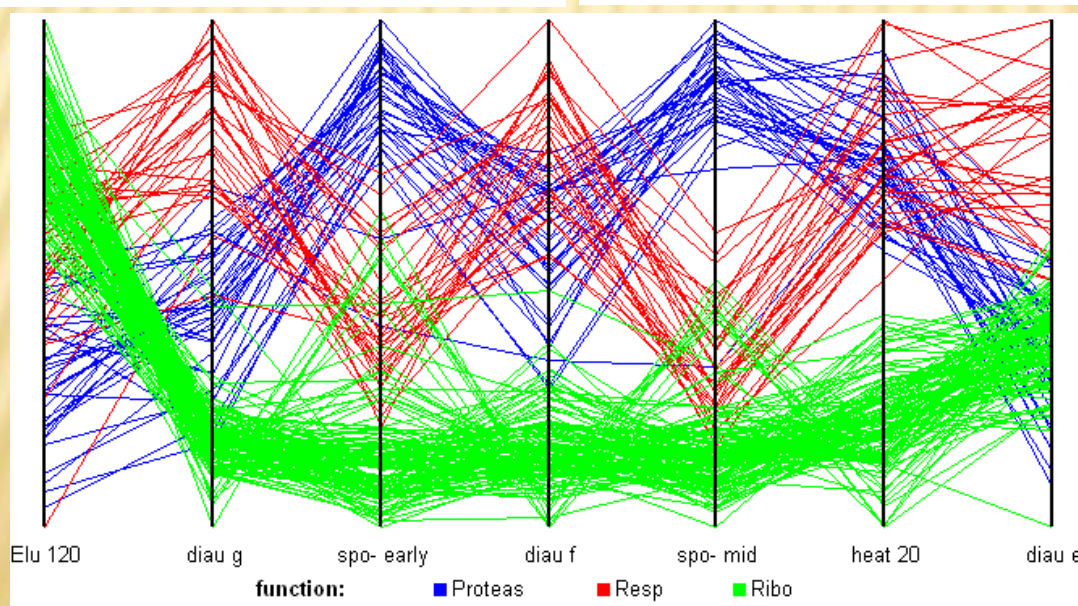
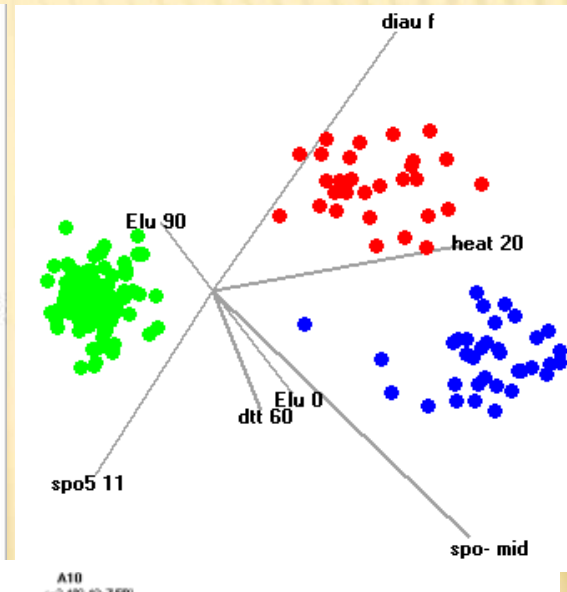
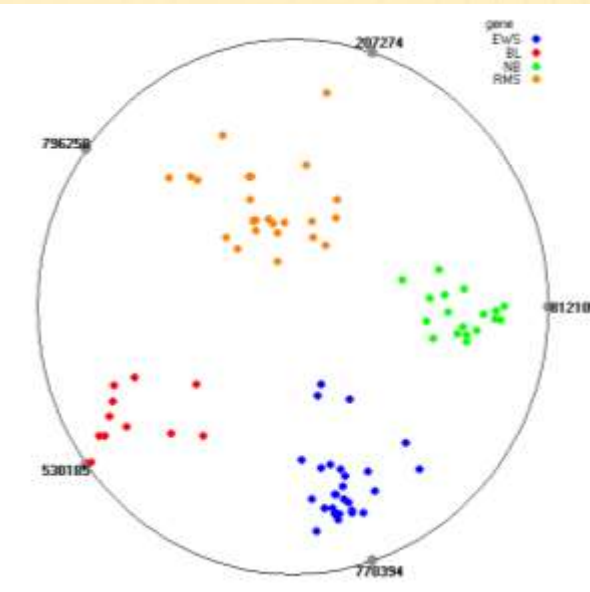
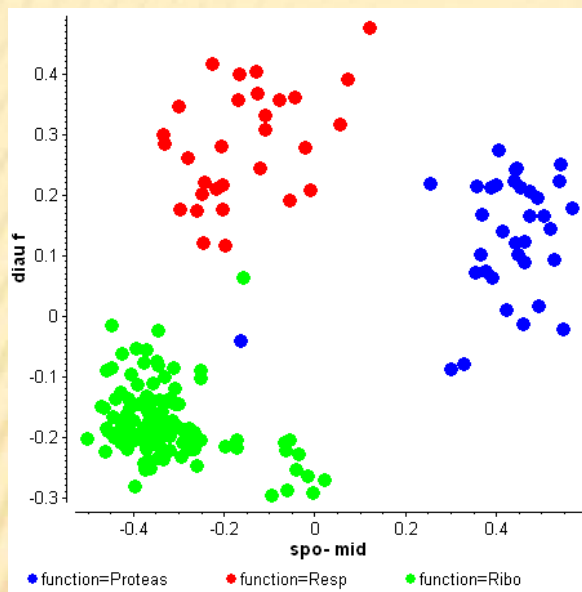
(9 EXAMPLES, 8 ATTRIBUTES)



VISUALIZATION: MDL



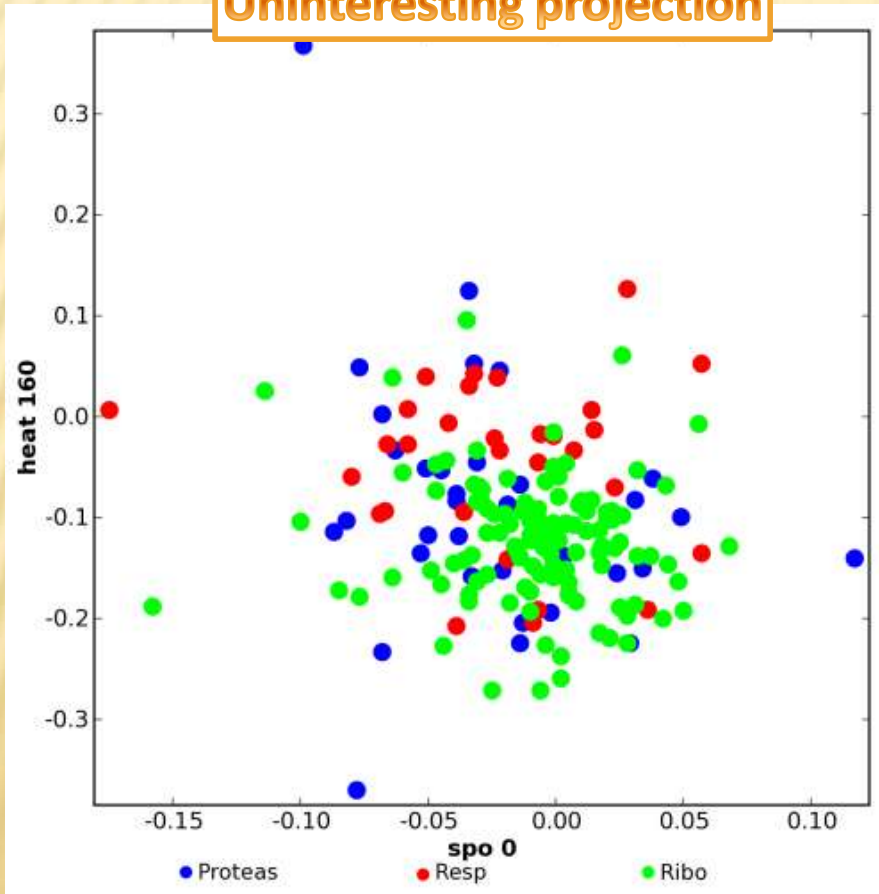
VISUALIZATION METHODS: MORE ATTS



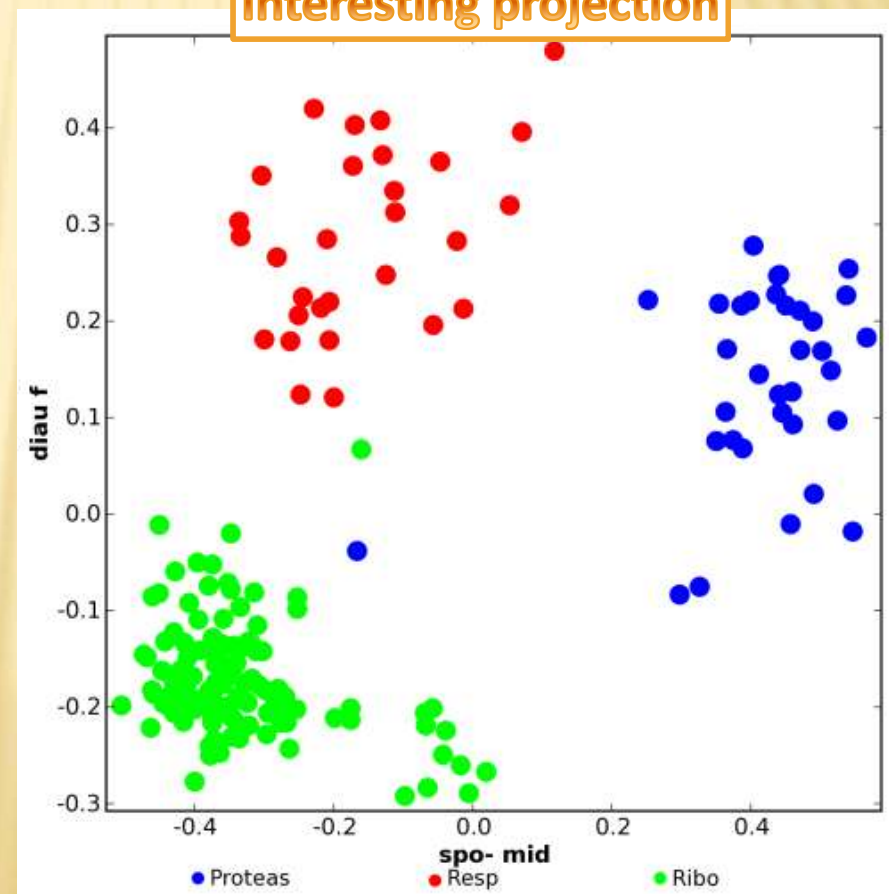
POINT-BASED METHODS

- ✗ Scatterplot, radviz, linear projections, ...
- ✗ Projection is more interesting if it separates points from different classes

Uninteresting projection

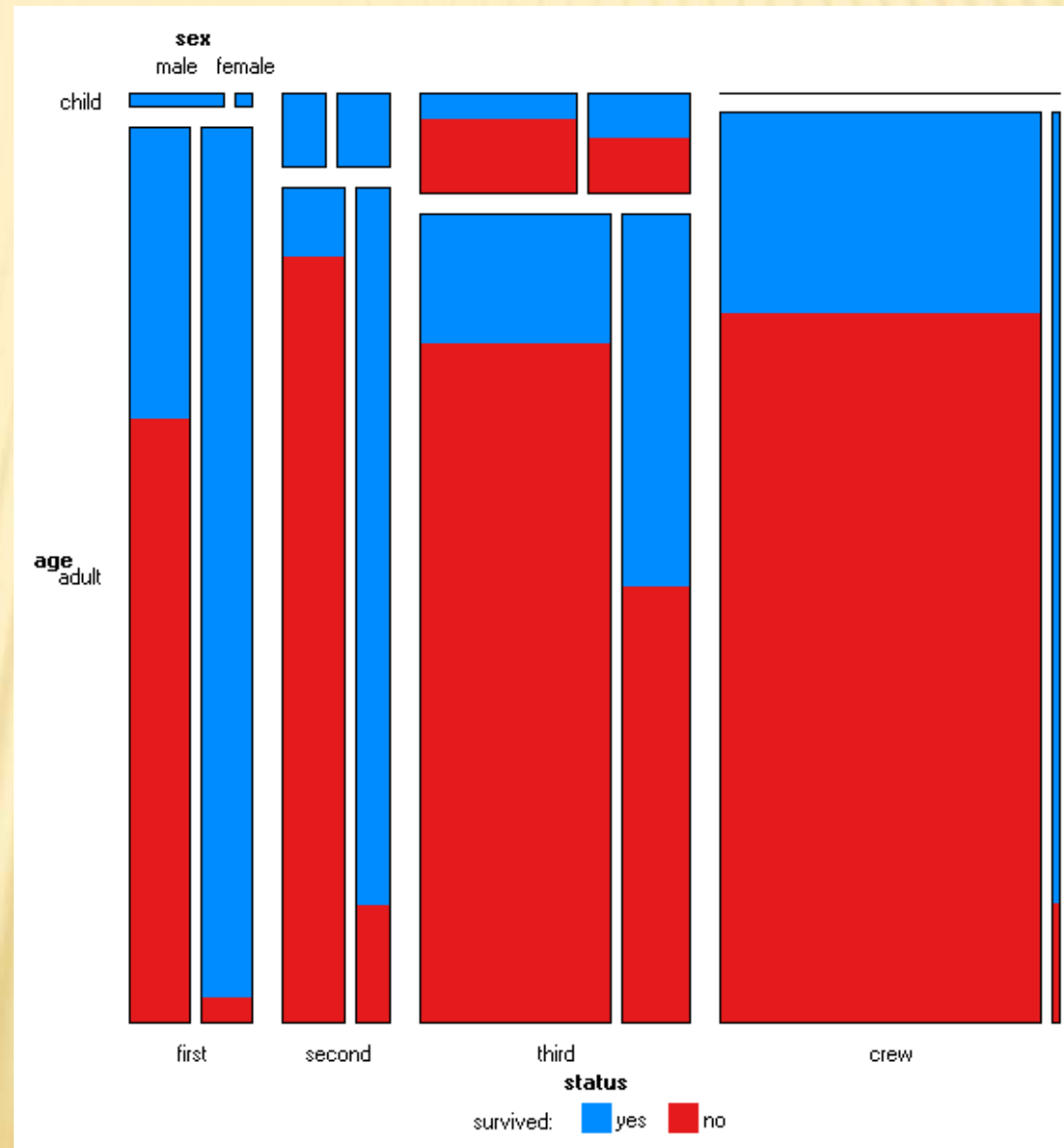


Interesting projection



MOSAIC PLOT

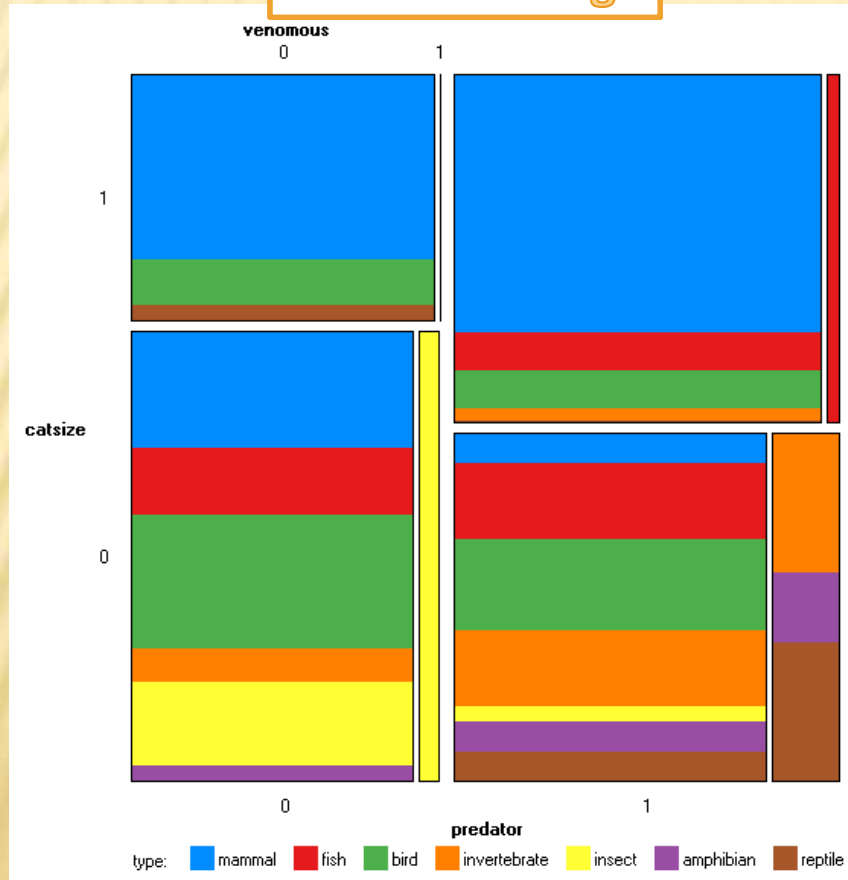
- ✗ Visualization of discrete attributes
- ✗ Recursive cell splitting according to attribute values
- ✗ Color = class distribution
- ✗ Interestingness depends on the selection of attributes and their positions



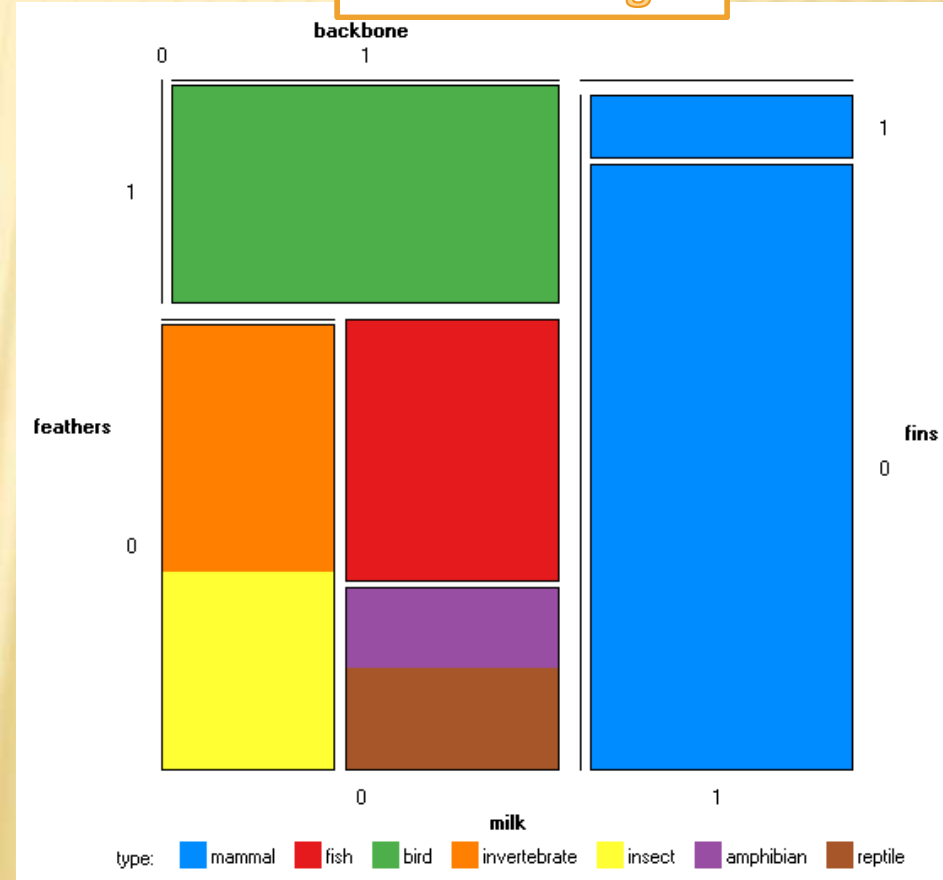
SELECTION OF ATTRIBUTES

- ✗ Attributes influence the “purity” of cells

Uninteresting



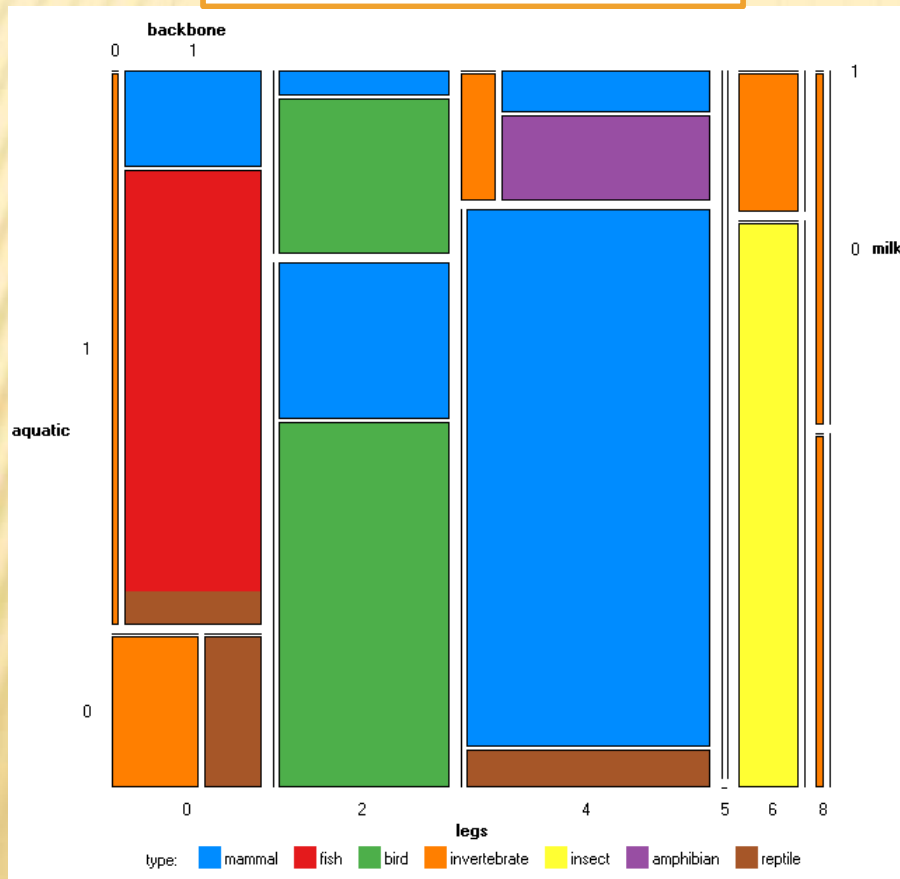
Interesting



ORDER OF ATTRIBUTES

- ✗ Order influences the ability to see the rules that include more than one cell:

Uninteresting order



Interesting order

