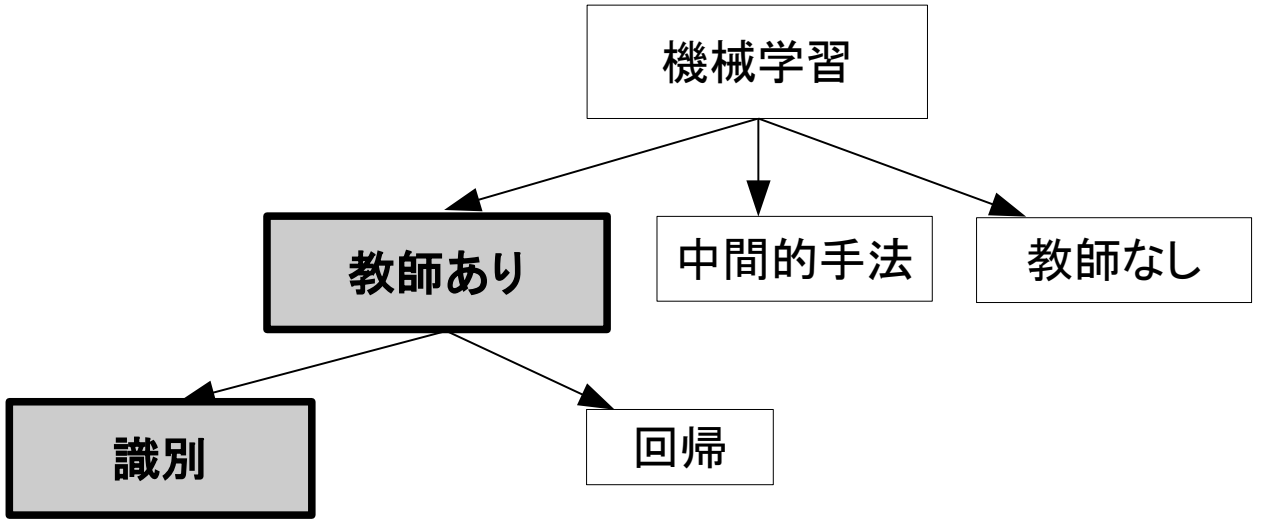
基本的な教師あり学習

- 決定木
- ナイーブベイズ
- ロジスティック識別

3. 識別 ー概念学習と決定木ー

- 問題設定
  - 教師あり学習
  - ラベル入力　→　ラベル出力



- ラベル特徴

年齢 = 若年
処方 = 近視
乱視 = なし
涙量 = 正常

⇒ 推薦レンズ = ソフト

- 数値特徴

contact-lenses データ

Viewer

Relation: contact-lenses

	1: age Nominal	2: spectacle-prescrip Nominal	3: astigmatism Nominal	4: tear-prod-rate Nominal	5: contact-lenses Nominal
1	young	myope	no	reduced	none
2	young	myope	no	normal	soft
3	young	myope	yes	reduced	none
4	young	myope	yes	normal	hard
5	young	hypermetrope	no	reduced	none
6	young	hypermetrope	no	normal	soft
7	young	hypermetrope	yes	reduced	none
8	young	hypermetrope	yes	normal	hard
9	pre-presbyopic	myope	no	reduced	none
10	pre-presbyopic	myope	no	normal	soft
11	pre-presbyopic	myope	yes	reduced	none
12	pre-presbyopic	myope	yes	normal	hard
13	pre-presbyopic	hypermetrope	no	reduced	none
14	pre-presbyopic	hypermetrope	no	normal	soft
15	pre-presbyopic	hypermetrope	yes	reduced	none
16	pre-presbyopic	hypermetrope	yes	normal	none
17	presbyopic	myope	no	reduced	none
18	presbyopic	myope	no	normal	none
19	presbyopic	myope	yes	reduced	none
20	presbyopic	myope	yes	normal	hard
21	presbyopic	hypermetrope	no	reduced	none
22	presbyopic	hypermetrope	no	normal	soft
23	presbyopic	hypermetrope	yes	reduced	none
24	presbyopic	hypermetrope	yes	normal	none

Undo OK Cancel

年齢・処方・
乱視・涙量

推薦コンタクトレンズ
none, soft, hard

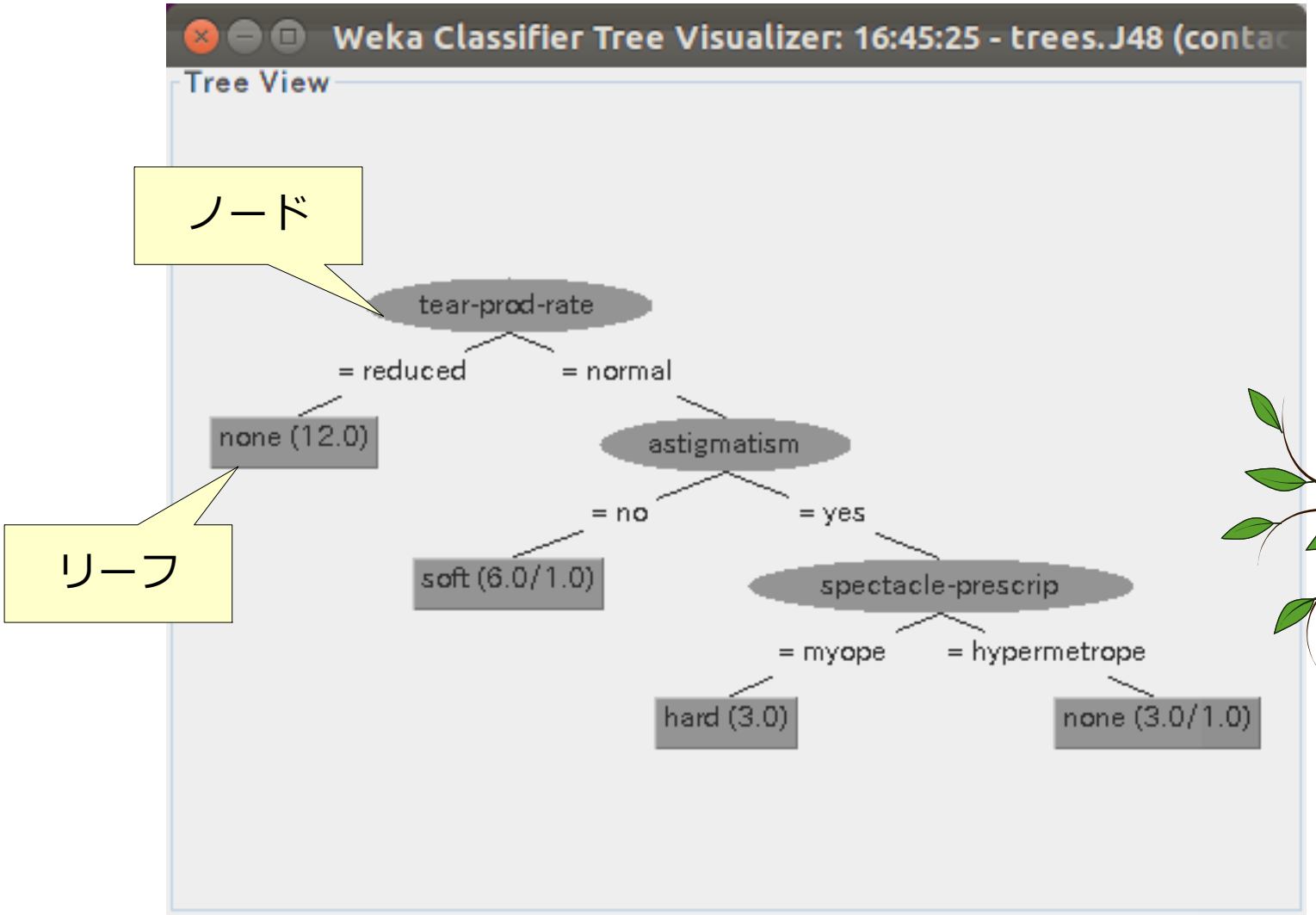
3.2 概念学習とは

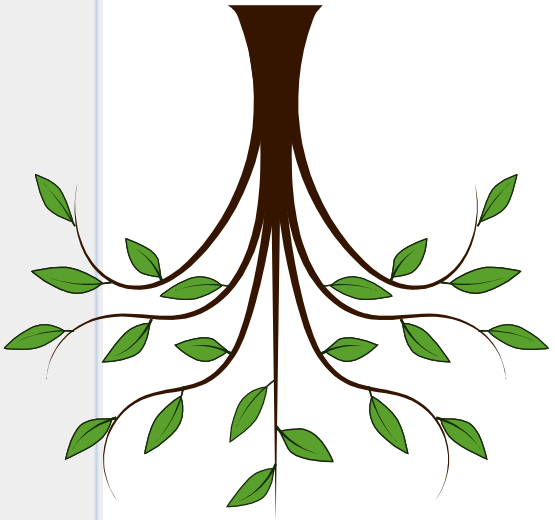
- 概念学習とは
  - 正解の概念を説明する特徴ベクトルの性質（論理式）を求めること
  - 論理式の例

    （乱視 = あり） ∧ （ドライアイ = なし）⇒ soft
- 学習の方法
  - 可能な論理式が少数
    - 正解概念の候補を絞り込んでゆく（候補削除アルゴリズム）
  - 可能な論理式が多数
    - バイアス（偏見）をかけて探索する ⇨ 決定木

3.4 決定木の学習

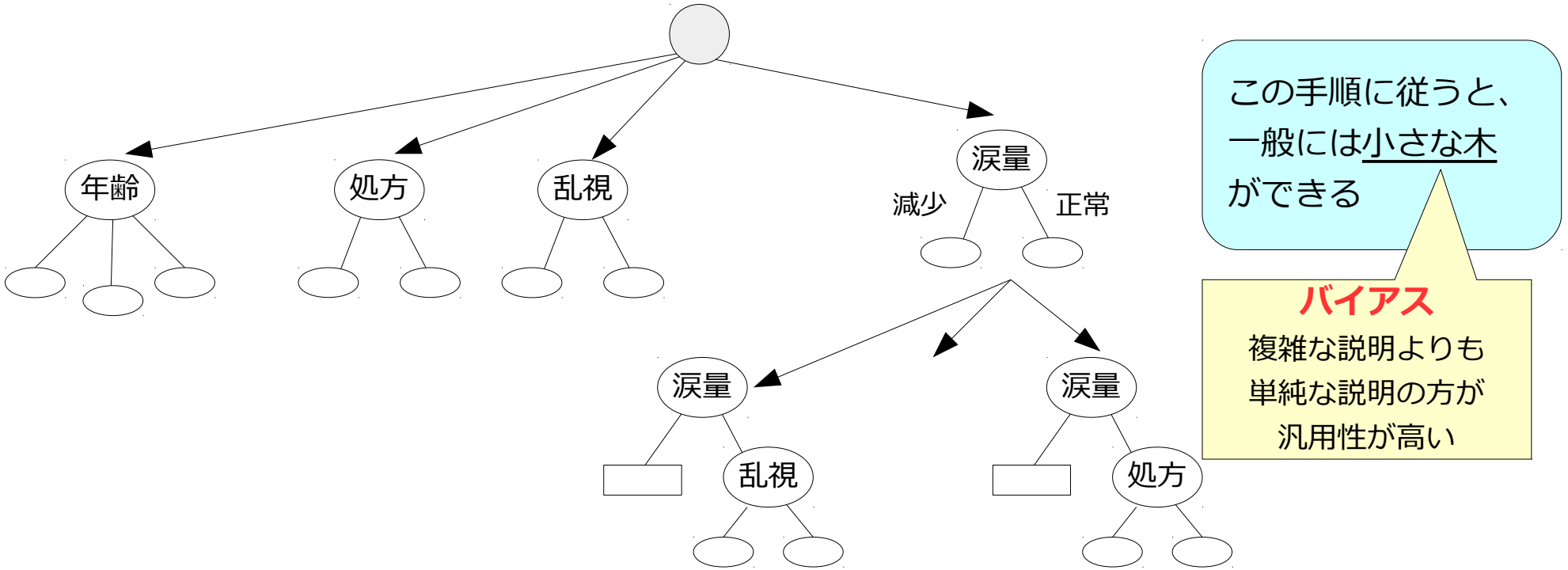
- 学習した決定木の例





3.4 決定木の学習

- 決定木学習の考え方
  - ノードは、データを分割する条件を持つ
    - できるだけ同一クラスのデータがリーフに偏るように
  - 分割後のデータ集合に対して、同様の操作を行う
  - 全てのリーフが単一クラスの集合になれば終了



この手順に従うと、
一般には小さな木
ができる

バイアス
複雑な説明よりも
単純な説明の方が
汎用性が高い

属性の分類能力 (1/2)

- 分類能力の高い属性を決定する方法
  - その属性を使った分類を行うことによって、なるべくきれいにクラスが分かれるように
  - エントロピー
    - データ集合 S の乱雑さを表現
    - 正例の割合：p^+ ， 負例の割合：p^-
    - エントロピーの定義

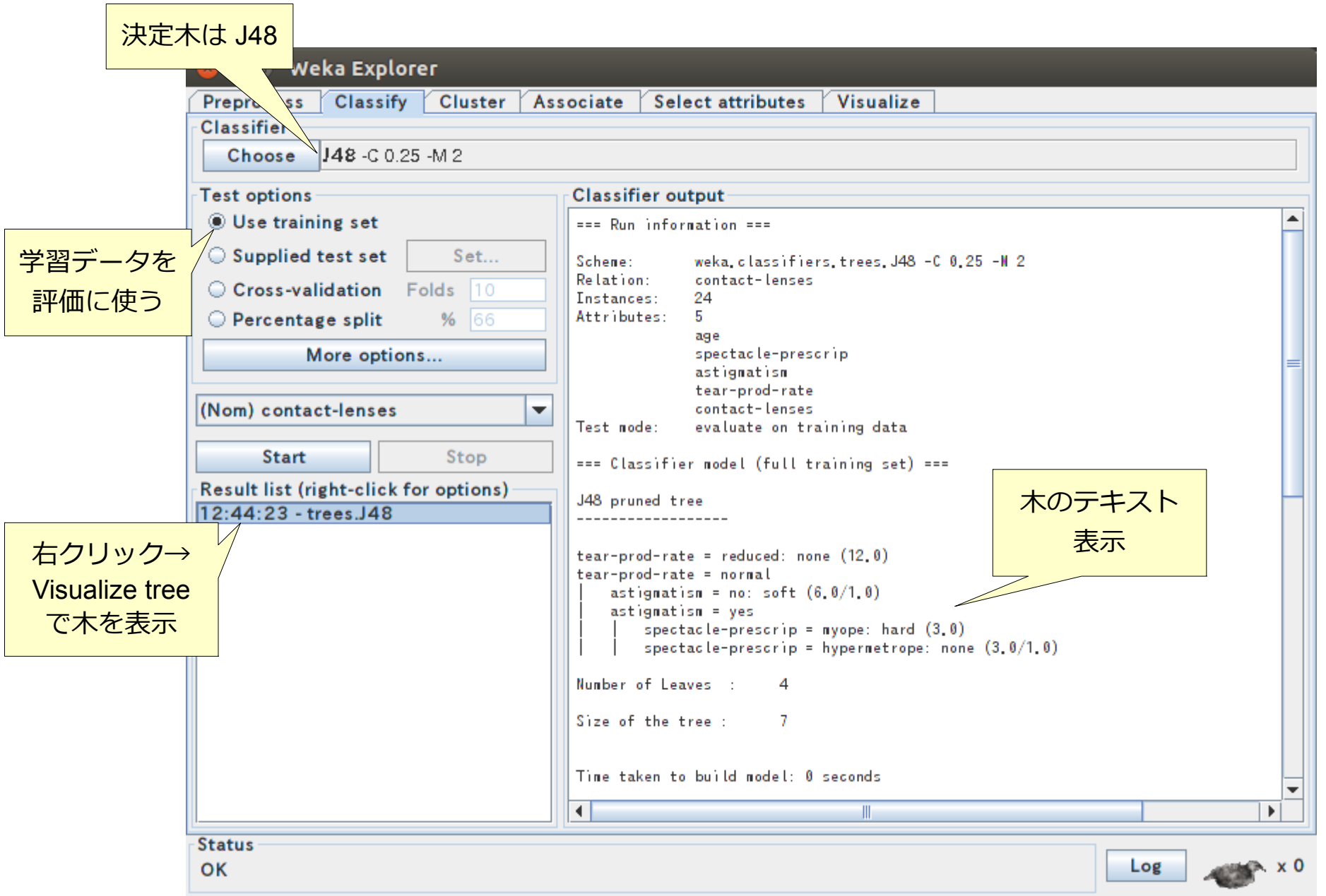
$$Entropy(S) = -p^+ \log p^+ - p^- \log p^-$$

属性の分類能力 (2/2)

- 情報獲得量
  - 属性 A を用いた分類後のエントロピーの減少量
  - 値 v を取る訓練例の集合：Sv
  - Sv の要素数：|Sv|
  - 情報獲得量の定義

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|Sv|}{|S|} Entropy(Sv)$$

Weka での決定木学習

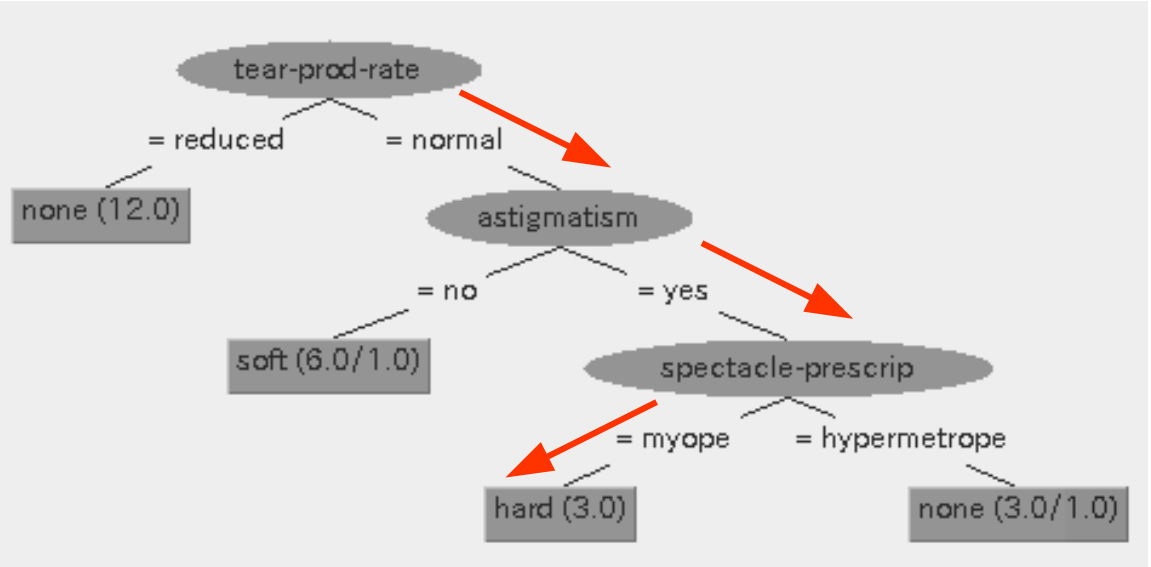


実行例

- 入力

  年齢　処方　乱視　涙量

  (young, myope, yes, normal)



- 出力

  hard

バイアスの検討

なぜ単純な木の方がよいか

- オッカムの剃刀

  「データに適合する最も単純な仮説を選べ」

  - 複雑な仮説

    → 表現能力が高い

    → 偶然にデータを説明できるかもしれない

  - 単純な仮説

    → 表現能力が低い

    → 偶然にデータを説明できる確率は低い

    → でも説明できた！

    → **必然**

連続値属性の扱い

- 連続値 A を持つ属性から真偽値 (A < c?) を値とするノードを作成

→c をどうやって決めるか

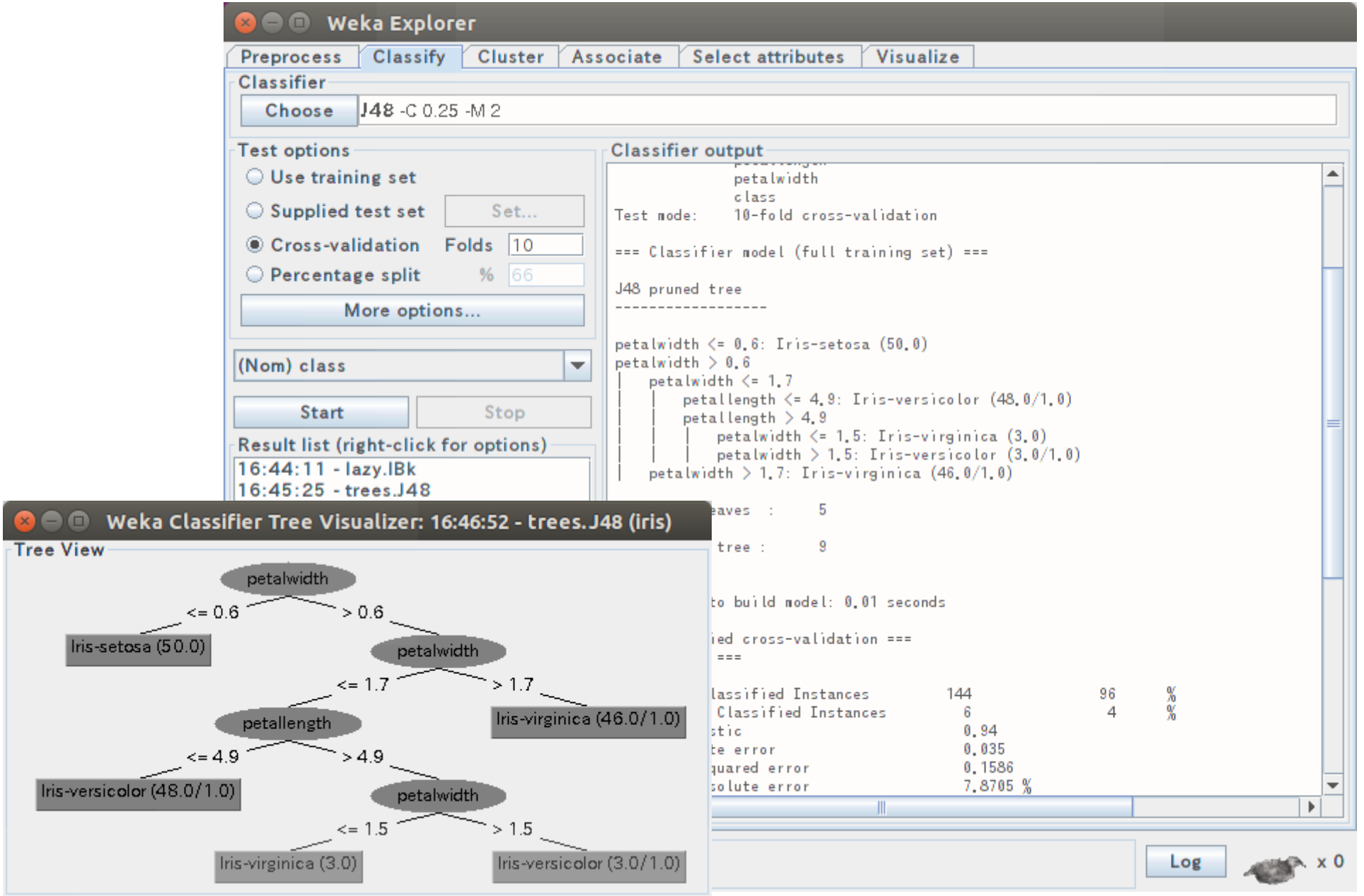
気温	40	48	60	72	80	90
playTennis	No	No	Yes	Yes	Yes	No

c=(48+60)/2
=54

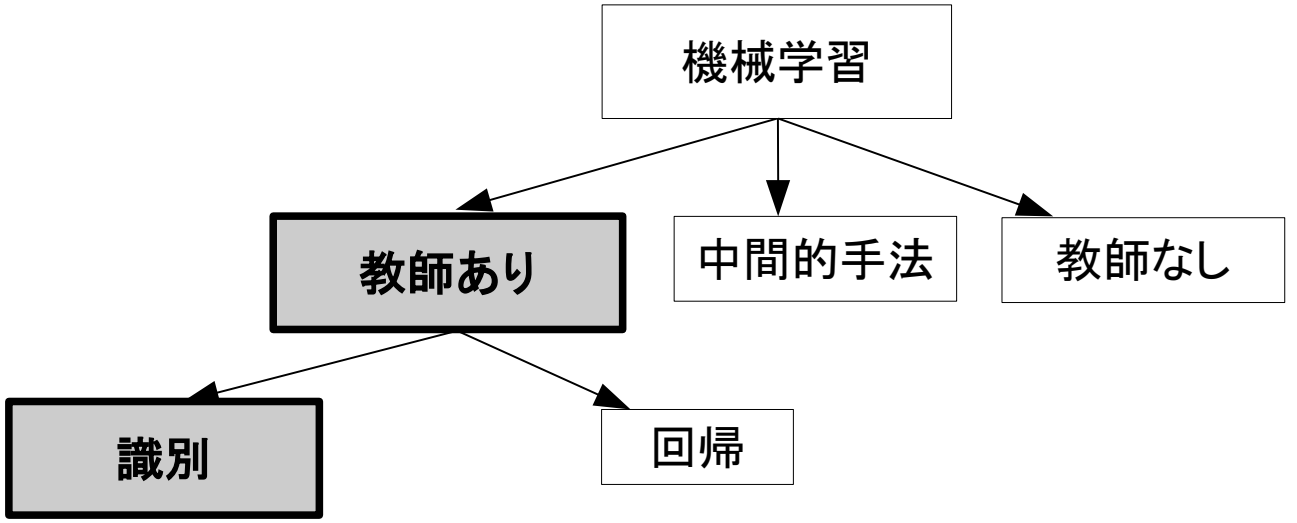
c=(80+90)/2
=85

情報獲得量の多い方

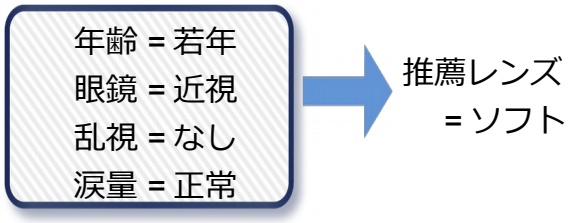
連続値属性の扱い


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
16:44:11 - lazy.IBk
16:45:25 - trees.J48
Classifier output
petalwidth
class
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
J48 pruned tree
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
| petalwidth <= 1.7
| | petallength <= 4.9: Iris-versicolor (48.0/1.0)
| | petallength > 4.9
| | | petalwidth <= 1.5: Iris-virginica (3.0)
| | | petalwidth > 1.5: Iris-versicolor (3.0/1.0)
| petalwidth > 1.7: Iris-virginica (46.0/1.0)
Log
x 0
Weka Classifier Tree Visualizer: 16:46:52 - trees.J48 (iris)
Tree View
petalwidth
<= 0.6
> 0.6
Iris-setosa (50.0)
petalwidth
<= 1.7
> 1.7
petallength
Iris-virginica (46.0/1.0)
<= 4.9
> 4.9
Iris-versicolor (48.0/1.0)
petalwidth
<= 1.5
> 1.5
Iris-virginica (3.0)
Iris-versicolor (3.0/1.0)

4. 識別　―統計的手法―



- ラベル特徴



- 数値特徴

4.1 統計的識別とは

- 最大事後確率則による識別

$$C_{MAP} = \arg\max_i P(\omega_i|\boldsymbol{x})$$

$\boldsymbol{x}$ ：特徴ベクトル

$\omega_i \quad (1 \leq i \leq c)$：クラス

  - データから直接的にこの確率を求めるのは難しい

- ベイズの定理　$P(A|B) = \dfrac{P(B|A)P(A)}{P(B)}$

$$\begin{aligned} C_{MAP} &= \arg\max_i P(\omega_i|\boldsymbol{x}) \\ &= \arg\max_i \frac{P(\boldsymbol{x}|\omega_i)P(\omega_i)}{P(\boldsymbol{x})} \\ &= \arg\max_i P(\boldsymbol{x}|\omega_i)P(\omega_i) \end{aligned}$$

4.1 統計的識別とは

- 事前確率

  $\mathrm{P}(\omega_i)$

  - 特徴ベクトルを観測する前の、各クラスの起こりやすさ

- 事前確率の最尤推定

$$P(\omega_i) = \frac{n_i}{N}$$

N: 全データ数、n_i: クラス i のデータ数

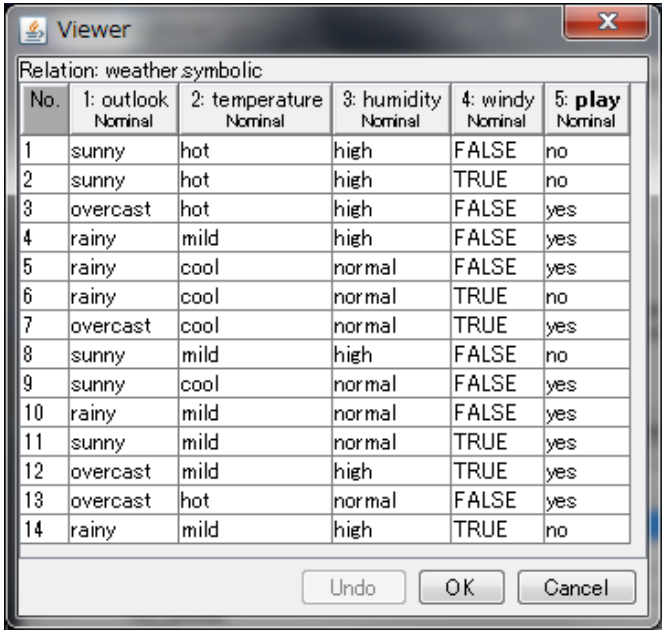
4.1 統計的識別とは

- 尤度

  $\mathrm{P}(\boldsymbol{x}|\omega_i)$

  - 特定のクラスから、ある特徴ベクトルが出現する尤もらしさ

- d 次元ベクトルの場合の最尤推定
  - 値の組合せがデータ中に出現しないもの多数



Weka の weather.nominal データ

3×3×2×2=36 種類の組合せ

4.2.2 ナイーブベイス識別

- ナイーブベイズの近似
  - 全ての特徴が独立であると仮定

$$P(\boldsymbol{x}|\omega_i) = P(x_1, \ldots, x_d|\omega_i)$$

$$= \prod_{j=1}^{d} P(x_j|\omega_i)$$

$$C_{NB} = \arg\max_i P(\omega_i) \prod_{j=1}^{d} P(x_j|\omega_i)$$

4.2.2 ナイーブベイス識別

- 尤度の最尤推定

$$P(x_j|\omega_i) = \frac{n_{ij}}{n_i}$$

n_{ij}: クラス i のデータのうち、
j 次元目の値が x_j の個数

ゼロ頻度問題

- 確率の m 推定

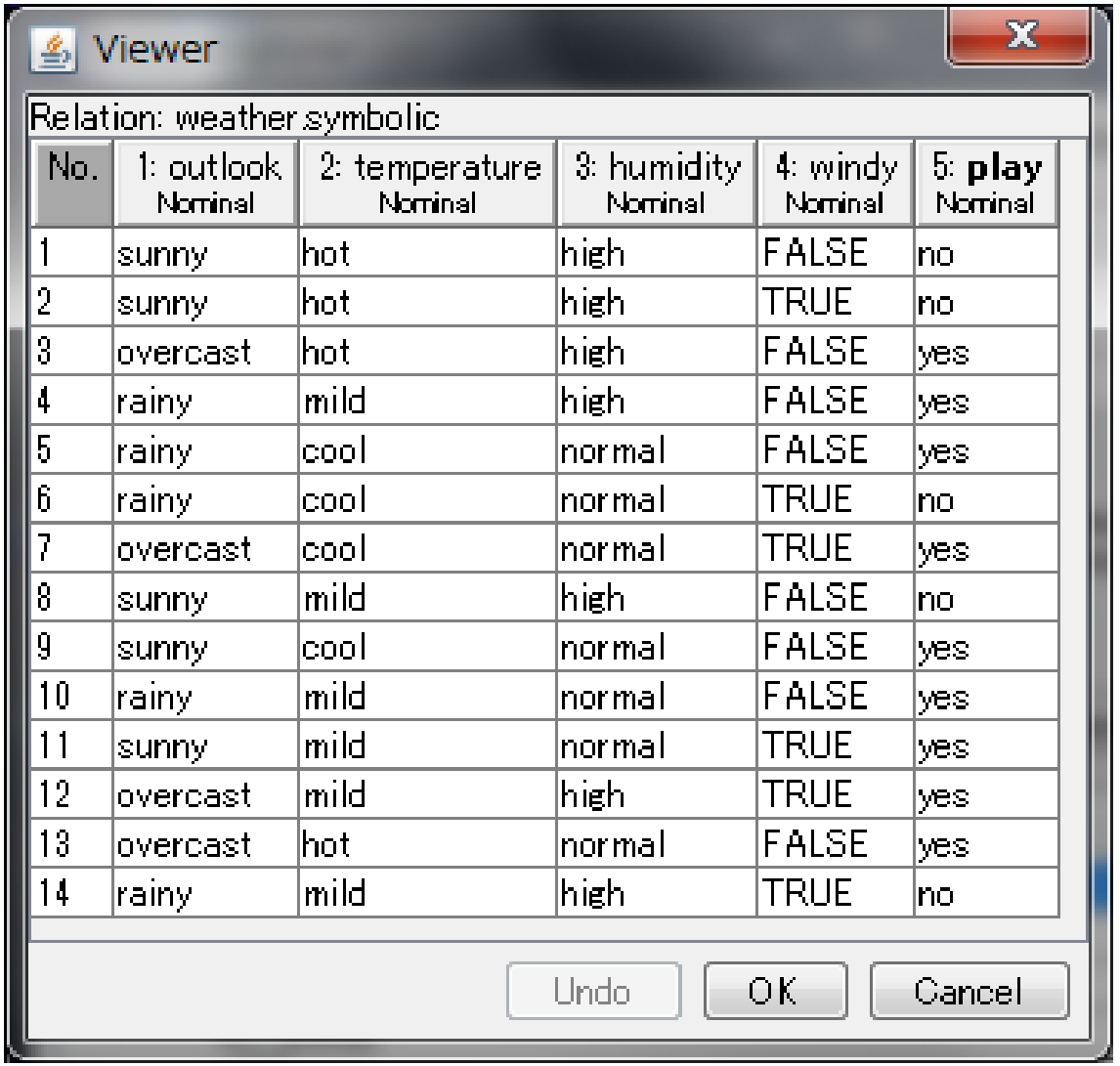
$$P(x_j|\omega_i) = \frac{n_{ij} + mp}{n_i + m}$$

p: 事前に見積もった各特徴値の割合
m: 事前に用意する標本数

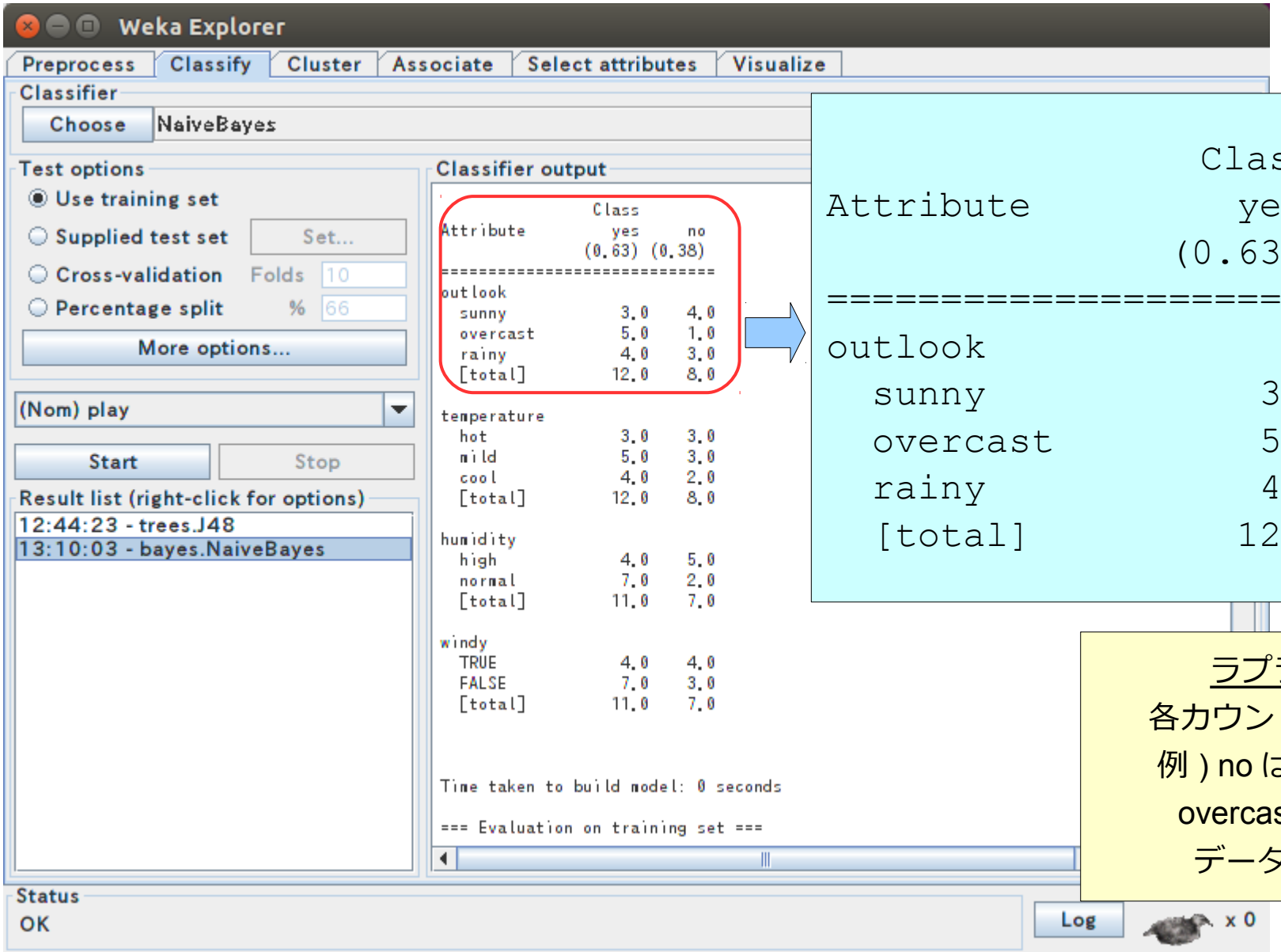
  - ラプラス推定
    - m: 特徴値の種類数、 p: 等確率　とすると、 *mp=1*

4.2.2 ナイーブベイス識別

weather.nominal データ



4.2.2 ナイーブベイス識別



Attribute	Class yes (0.63)	no (0.38)
outlook		
sunny	3.0	4.0
overcast	5.0	1.0
rainy	4.0	3.0
[total]	12.0	8.0

ラプラス推定

各カウントの初期値が 1

例) no は 5 事例で、overcast の事例はデータ中にない

実行例

- 入力

  天気　気温　湿度　　風

  x=(rainy, hot, high, false)

  $P(yes) = 0.63$

  $P(no) = 0.38$

  $P(\boldsymbol{x}|yes) = 4/12 \times 3/12 \times 4/11 \times 7/11 = 0.019$

  $P(\boldsymbol{x}|no) = 3/8 \times 3/8 \times 5/7 \times 3/7 = 0.043$

$$P(\boldsymbol{x}|yes) \cdot P(yes) = 0.012 < P(\boldsymbol{x}|no) \cdot P(no) = 0.016$$

- 出力

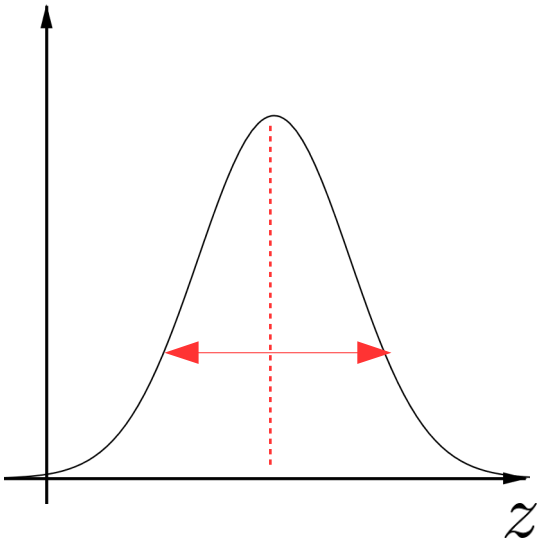
  no

5.2 数値特徴に対するベイズ識別
5.2.1 数値特徴に対するナイーブベイズ識別

$$C_{NB} = \arg\max_i P(\omega_i) \prod_{j=1}^{d} p(x_j|\omega_i)$$

- 確率密度関数 $p(x_j|\omega_i)$ の推定
  - 正規分布を仮定

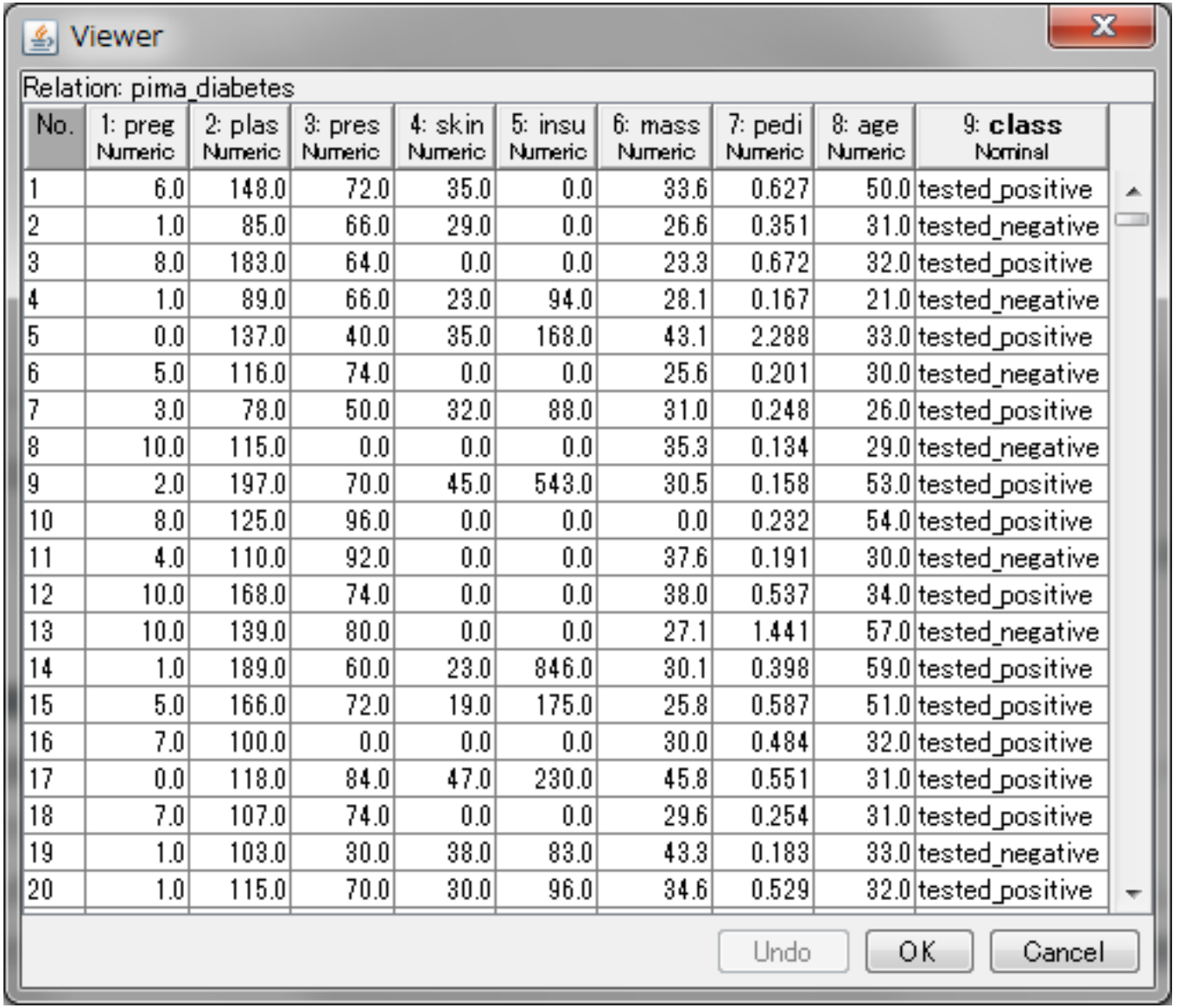
$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(z-\mu)^2}{2\sigma^2})$$



  - 平均 μ と分散 σ を最尤推定
    - それぞれ、学習データの平均と分散になる

5.2.1 数値特徴に対するナイーブベイズ識別

diabetes データ



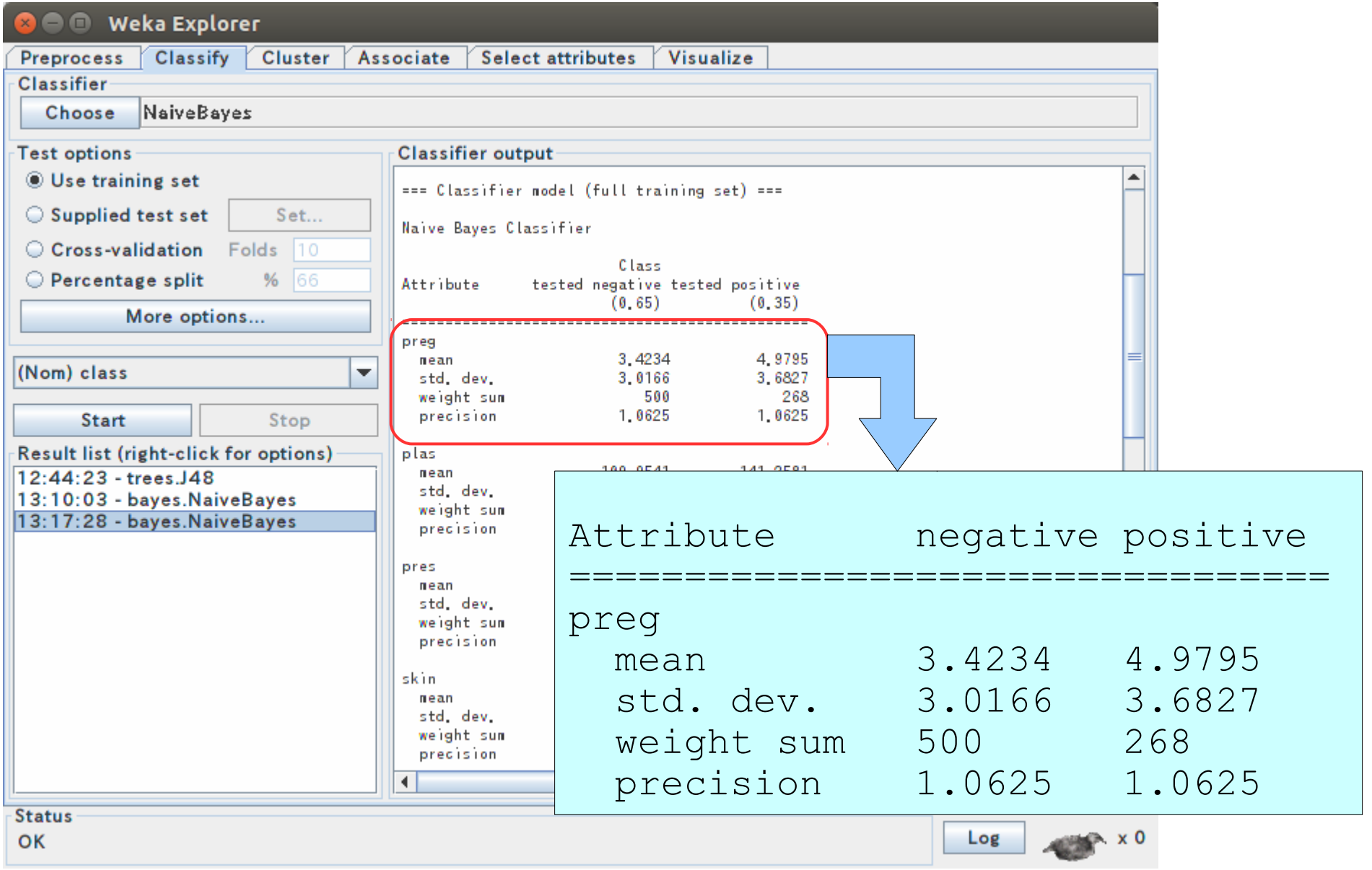
Viewer

Relation: pima_diabetes

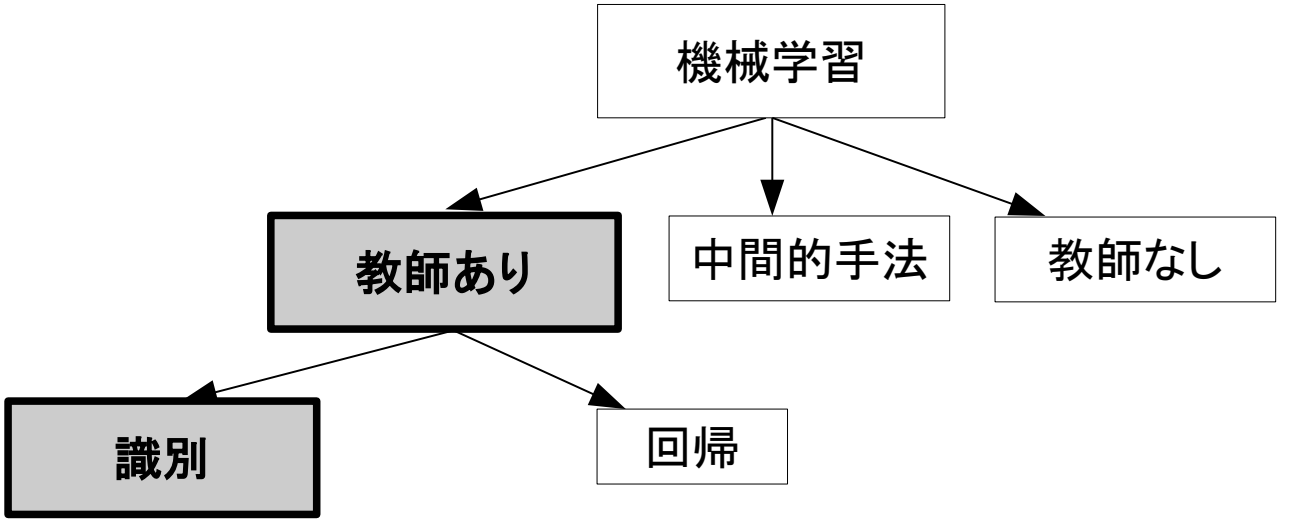
No.	1: preg Numeric	2: plas Numeric	3: pres Numeric	4: skin Numeric	5: insu Numeric	6: mass Numeric	7: pedi Numeric	8: age Numeric	9: **class** Nominal
1	6.0	148.0	72.0	35.0	0.0	33.6	0.627	50.0	tested_positive
2	1.0	85.0	66.0	29.0	0.0	26.6	0.351	31.0	tested_negative
3	8.0	183.0	64.0	0.0	0.0	23.3	0.672	32.0	tested_positive
4	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21.0	tested_negative
5	0.0	137.0	40.0	35.0	168.0	43.1	2.288	33.0	tested_positive
6	5.0	116.0	74.0	0.0	0.0	25.6	0.201	30.0	tested_negative
7	3.0	78.0	50.0	32.0	88.0	31.0	0.248	26.0	tested_positive
8	10.0	115.0	0.0	0.0	0.0	35.3	0.134	29.0	tested_negative
9	2.0	197.0	70.0	45.0	543.0	30.5	0.158	53.0	tested_positive
10	8.0	125.0	96.0	0.0	0.0	0.0	0.232	54.0	tested_positive
11	4.0	110.0	92.0	0.0	0.0	37.6	0.191	30.0	tested_negative
12	10.0	168.0	74.0	0.0	0.0	38.0	0.537	34.0	tested_positive
13	10.0	139.0	80.0	0.0	0.0	27.1	1.441	57.0	tested_negative
14	1.0	189.0	60.0	23.0	846.0	30.1	0.398	59.0	tested_positive
15	5.0	166.0	72.0	19.0	175.0	25.8	0.587	51.0	tested_positive
16	7.0	100.0	0.0	0.0	0.0	30.0	0.484	32.0	tested_positive
17	0.0	118.0	84.0	47.0	230.0	45.8	0.551	31.0	tested_positive
18	7.0	107.0	74.0	0.0	0.0	29.6	0.254	31.0	tested_positive
19	1.0	103.0	30.0	38.0	83.0	43.3	0.183	33.0	tested_negative
20	1.0	115.0	70.0	30.0	96.0	34.6	0.529	32.0	tested_positive

Undo OK Cancel

5.2.1 数値特徴に対するナイーブベイズ識別



5. 識別　ー生成モデルと識別モデルー

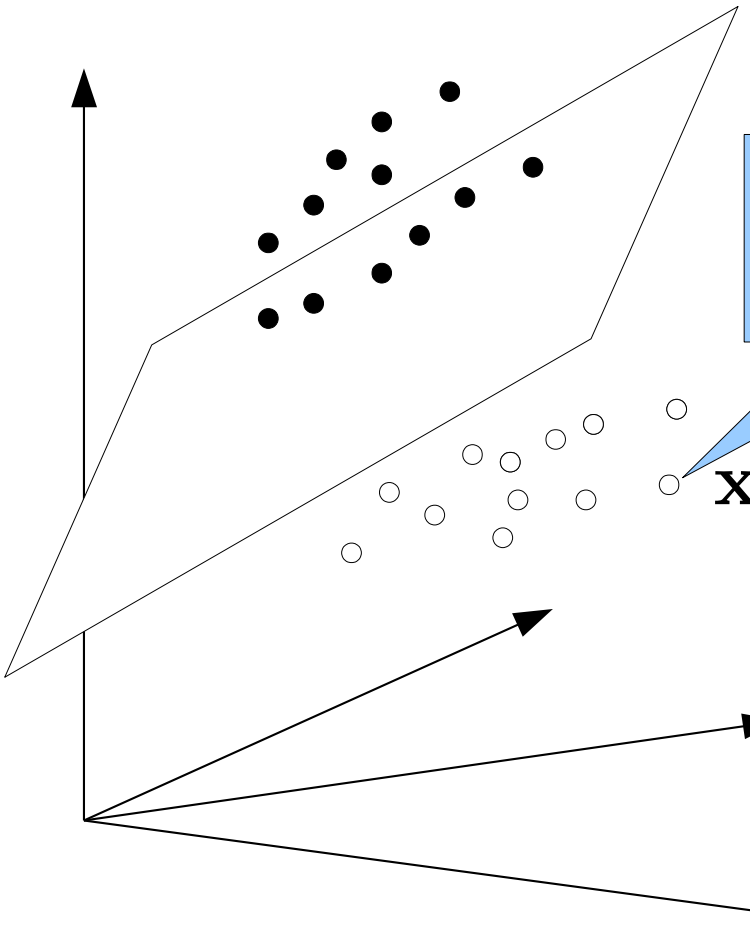


- ラベル特徴
- 数値特徴

気温 = 20
湿度 = 60
降水確率 = 0

Play = Yes

5.1 数値特徴に対する「教師あり・識別」問題の定義



特徴ベクトルは、
特徴空間上の
点として表現される

$$\mathbf{x} = (x_1, \ldots, x_d)^T$$

比較的クラスが分離している
→ 小数のパラメータで各クラスを表現可能
⇒ **統計的手法**

クラス境界が複雑
非線形識別面 ⇒ **ニューラルネット**
高次元へマッピング⇒ **SVM**

5.2.2 生成モデルの考え方

- 事後確率を求めるにあたって、同時確率を求めている
  - データが生成される様子をモデル化していると見ることも出来る
    - 事前確率に基づいてクラスを選ぶ
    - そのもとで、特徴ベクトルを出力する

$$P(\omega_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x})}$$

$$= \frac{p(\omega_i, \boldsymbol{x})}{p(\boldsymbol{x})}$$

事後確率を求めるより、難しい問題を解いているのではないか？

5.3.1 識別モデルの考え方

- 事後確率を直接求める

正例で
大きな値

負例で
大きな値

あまり
関係ない

$$\boldsymbol{x} = (x_1, \quad x_2, \quad x_3, \quad \ldots, \quad x_d)$$

w_1 正の 大きな値

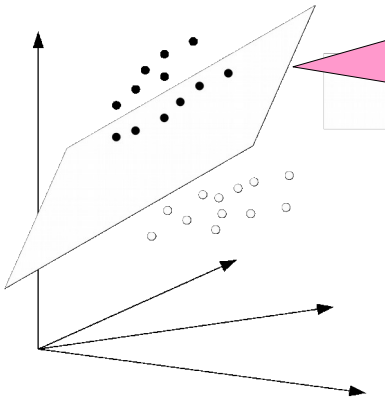
w_2 負の 大きな値

w_3 小さな値

正例／負例
の閾値

$$w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots + w_d x_d + w_0$$

この値が正なら正例、
負なら負例となるように
重みｗを学習する



この平面を
求めている
ことになる

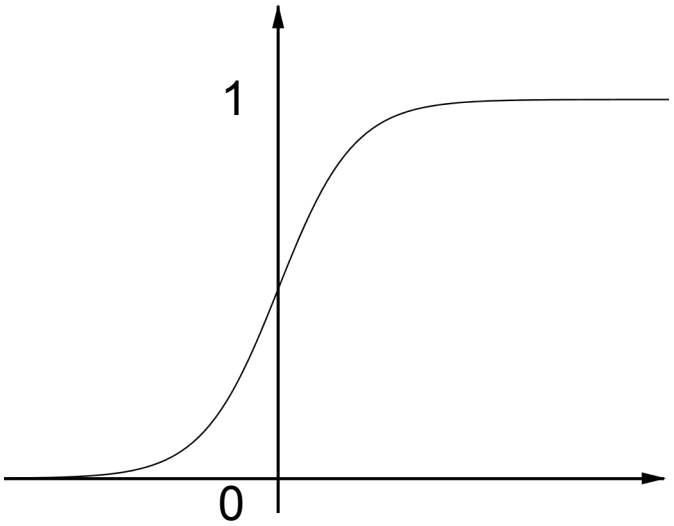
確率と対応づけるには？

5.3.1 識別モデルの考え方

- ロジスティック識別
  - 入力が正例である確率

$$P(\oplus|\boldsymbol{x}) = \frac{1}{1+\exp(-(\boldsymbol{w}\cdot\boldsymbol{x}+w_0))}$$

-∞ ～ +∞ の値域を持つものを、順序を変えずに 0 ～ 1 にマッピング



シグモイド関数

5.3.2 ロジスティック識別器の学習

- 最適化対象＝モデルが学習データを生成する確率

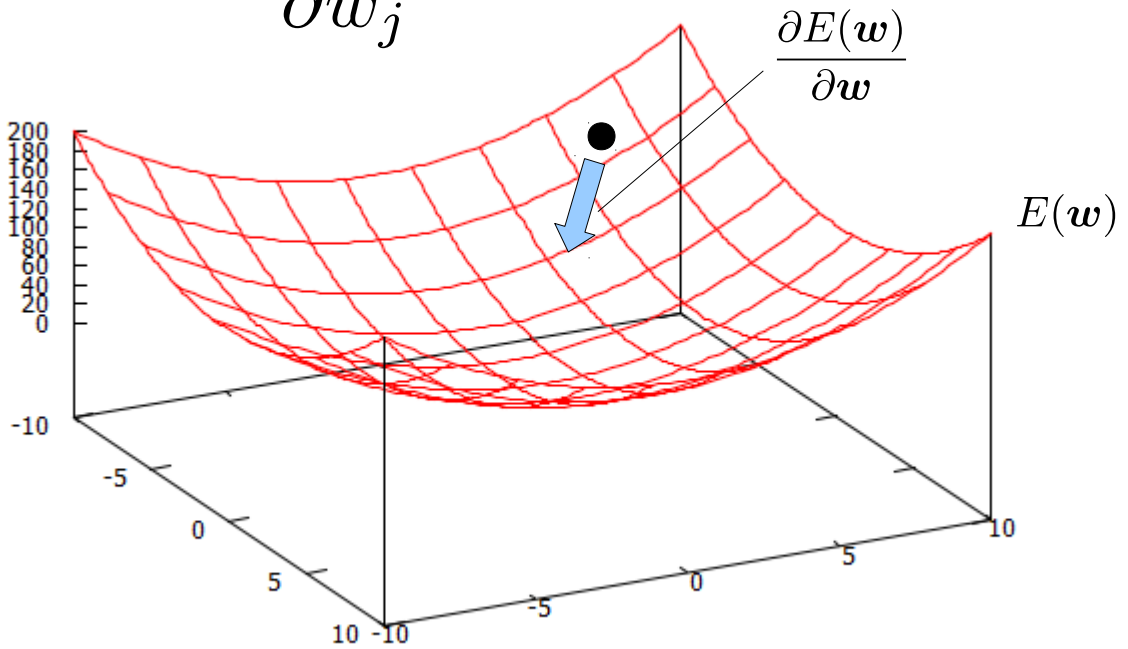
$$E(\boldsymbol{w}) = -\log P(D|\boldsymbol{w}) = -\log \prod_{\boldsymbol{x}_i \in D} o_i^{y_i} (1-o_i)^{(1-y_i)}$$

  - $E(\boldsymbol{w})$ を最急勾配法で最小化

$$w_j \leftarrow w_j - \eta \frac{\partial E(\boldsymbol{w})}{\partial w_j}$$

$$o = P(\oplus|\boldsymbol{x})$$
$$y = o \ \ or \ \ 1$$

正解ラベル



5.3.2 ロジスティック識別器の学習

- 重み更新量の計算

$$\frac{\partial E(\boldsymbol{w})}{\partial w_j} = \sum_{\boldsymbol{x}_i \in D} \left(\frac{y_i}{o_i} - \frac{1-y_i}{1-o_i}\right) o_i(1-o_i)x_{ij}$$

$$= \sum_{\boldsymbol{x}_i \in D} (y_i - o_i)x_{ij}$$

- 重みの更新式

$$w_j \leftarrow w_j - \eta \sum_{\boldsymbol{x}_i \in D} (y_i - o_i)x_{ij}$$

5.3.2 ロジスティック識別器の学習

diabetes データ

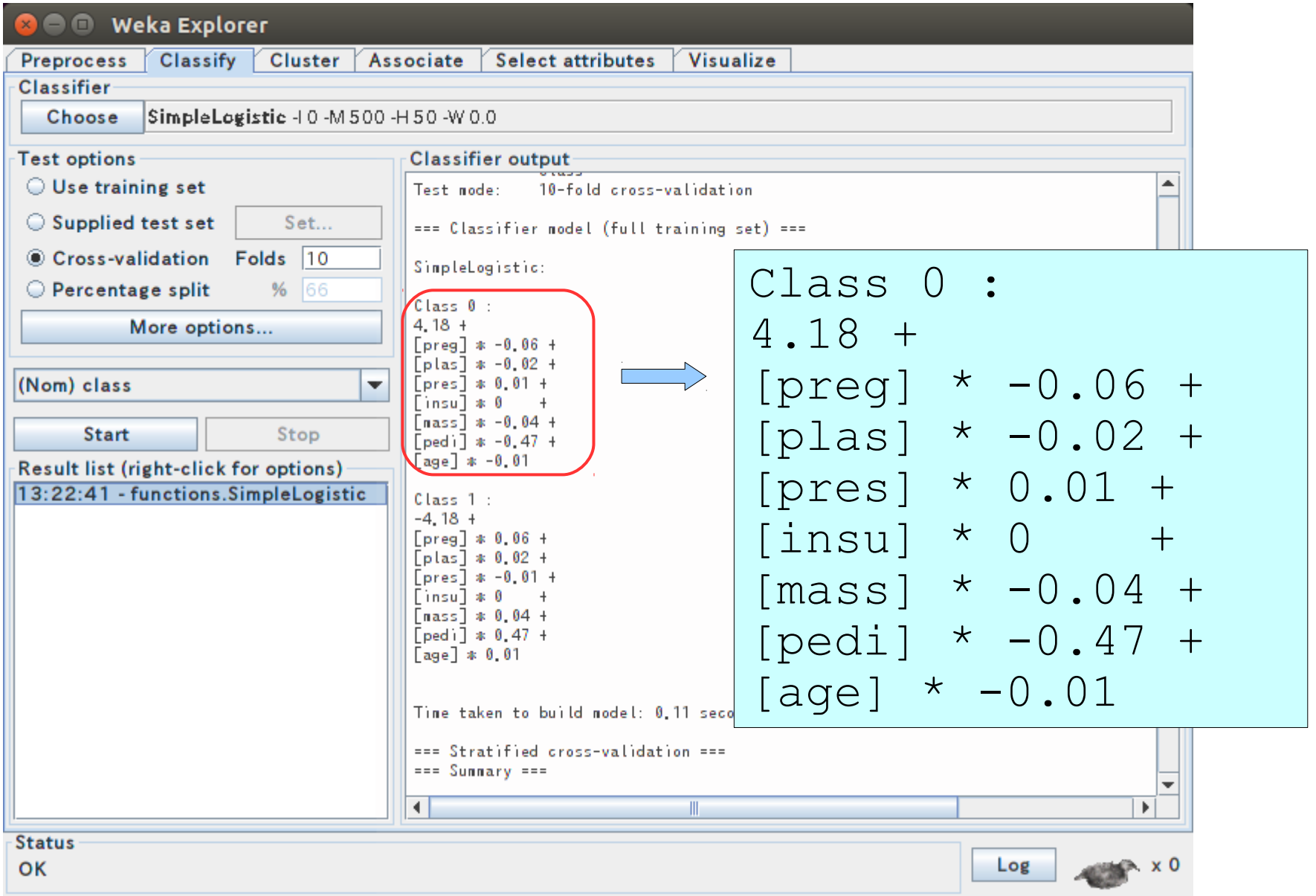
Viewer

Relation: pima_diabetes

No.	1: preg Numeric	2: plas Numeric	3: pres Numeric	4: skin Numeric	5: insu Numeric	6: mass Numeric	7: pedi Numeric	8: age Numeric	9: class Nominal
1	6.0	148.0	72.0	35.0	0.0	33.6	0.627	50.0	tested_positive
2	1.0	85.0	66.0	29.0	0.0	26.6	0.351	31.0	tested_negative
3	8.0	183.0	64.0	0.0	0.0	23.3	0.672	32.0	tested_positive
4	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21.0	tested_negative
5	0.0	137.0	40.0	35.0	168.0	43.1	2.288	33.0	tested_positive
6	5.0	116.0	74.0	0.0	0.0	25.6	0.201	30.0	tested_negative
7	3.0	78.0	50.0	32.0	88.0	31.0	0.248	26.0	tested_positive
8	10.0	115.0	0.0	0.0	0.0	35.3	0.134	29.0	tested_negative
9	2.0	197.0	70.0	45.0	543.0	30.5	0.158	53.0	tested_positive
10	8.0	125.0	96.0	0.0	0.0	0.0	0.232	54.0	tested_positive
11	4.0	110.0	92.0	0.0	0.0	37.6	0.191	30.0	tested_negative
12	10.0	168.0	74.0	0.0	0.0	38.0	0.537	34.0	tested_positive
13	10.0	139.0	80.0	0.0	0.0	27.1	1.441	57.0	tested_negative
14	1.0	189.0	60.0	23.0	846.0	30.1	0.398	59.0	tested_positive
15	5.0	166.0	72.0	19.0	175.0	25.8	0.587	51.0	tested_positive
16	7.0	100.0	0.0	0.0	0.0	30.0	0.484	32.0	tested_positive
17	0.0	118.0	84.0	47.0	230.0	45.8	0.551	31.0	tested_positive
18	7.0	107.0	74.0	0.0	0.0	29.6	0.254	31.0	tested_positive
19	1.0	103.0	30.0	38.0	83.0	43.3	0.183	33.0	tested_negative
20	1.0	115.0	70.0	30.0	96.0	34.6	0.529	32.0	tested_positive

Undo OK Cancel

5.3.2 ロジスティック識別器の学習




```
Class 0 :
4.18 +
[preg] * -0.06 +
[plas] * -0.02 +
[pres] * 0.01 +
[insu] * 0     +
[mass] * -0.04 +
[pedi] * -0.47 +
[age] * -0.01
```

実行例

- 入力

妊娠 血糖値 血圧 ...

x=(6, 148, 72, 35, 0, 33.6, 0.627, 50)

$$
\begin{aligned}
g(\boldsymbol{x}) &= 6 \times 0.06 + 148 \times 0.02 - 72 \times 0.01 \\
&\quad + 33.6 \times 0.04 + 0.627 \times 0.47 + 50 \times 0.01 \\
&\quad - 4.18 \\
&\quad = 0.559
\end{aligned}
$$

$$
P(tested\_positive) = 1/(1 + \exp(-g(\boldsymbol{x}))) = 0.636
$$

- 出力

tested_positive