APRENDIZAJE AUTOMÁTICO: TERCER CURSO DEL GRADO DE ING. INFORMÁTICA EN COMPUTACION

Clasificación:
Regresión Logística Binaria:
Aplicaciones con SPSS y Weka

César Hervás-Martínez
Grupo de Investigación AYRNA

Departamento de Informática y Análisis Numérico
Universidad de Córdoba
Campus de Rabanales. Edificio Einstein.
Email: chervas@uco.es

2019-2020

INDICE

1 Introducción

2 Modelo de regresión logística

3 Estimación de parámetros por máxima verosimilitud

4 Selección de variables predictoras

5 Conclusiones

INTRODUCCIÓN

Métodos de regresión

Describir relación entre una variable de respuesta (*dependiente*) Y, y variables explicativas (*independientes* o *predictoras*) $X_1, \ldots, X_k$

Regresión logística: método estándar para una clasificación binaria $Y \in \{0, 1\}$ (discreta en general)

Ejemplos: padecer o no una enfermedad, tener éxito o fracaso, comprar un producto o no comprarlo, evadir impuestos o no hacerlo, etc.

Introducción

En medicina, microbiología, y muchos otros campos es muy importante predecir el resultado de una variable de respuesta binaria (Aprendizaje supervisado).

El principal objetivo es aprender como distinguir ejemplos que pertenecen a una de entre dos clases (caracterizadas por los sucesos Y= 1, e Y= 0) en función de los valores que toman k variables predictoras o covariables.

Un modelo de regresión logística se puede representar de forma equivalente a como se representa una estructura de grafo de tipo perceptron con una función de activación logística.

INTRODUCCIÓN

Objetivos

1 Determinar la existencia o ausencia de relación entre una o más variables independientes y la variable dependiente binaria o multiclase.

2 Usar las variables independientes para predecir la probabilidad de que la variable de respuesta tome cada uno de sus dos o más posibles valores, en función de los valores de las variables independientes.

3 Utilizar estas probabilidades para clasificar observaciones futuras en una de las dos o más clases.

CONCEPTO / TIPOS

- **Regresión Logística: Técnica estadística multivariante → (Problemas de clasificación)**
  - **Explicar una variable dependiente discreta, asociada a la clase, en función de variables independientes (covariables) continuas o discretas.**
- **Según tipo variable discreta:**
  - **Variable dicotómica: 2 Clases → Regresión Logística Binaria**
  - **Variable multinomial: más de 2 Clases, sin relación entre si → Regresión Logística Multinomial**
    - **P. ej.: El tipo de suelo puede ser {Bosque, Industrial, Urbano}**
  - **Variable ordinal: más de 2 clases con un orden preestablecido → Regresión Logística Ordinal**
    - **P. ej.: La producción de un bien puede ser {Alta, Media, Baja}**

Modelos generalizados de regresión lineal: Regresión logística

Modelos lineales generalizados. (Hastie, and Tibshirani, 1990)

Regresión Logística (Cox and Snell, 1989; Hosmer and Lemeshow, 1989; Ryan, 1997)

Árboles de decisión y regresión logística (Landwehr, Hall, M., and Eibe, 2005)

Redes neuronales y regresión logística (Schumacher, Robner, and Vach,, 1996)

INTRODUCCIÓN

Si no es adecuada la regresión lineal, tenemos que construir un modelo no lineal.
Sea un modelo de regresión lineal múltiple de la forma

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + ... + \beta_k X_{ki} + \varepsilon_i \quad \text{para i=1,...,n}$$

siendo ε_i variables aleatorias indpendientes e idénticamente distribuidas

v.a.i.i.d. con distribución $N(0;\sigma^2)$

Si tratamos de aplicar este modelo al caso de que la v.a. *Y sea* dicotómica:

$$E(Y \mid X_1, X_2, ..., X_k) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + ... + \beta_k X_{ki} = p$$

Pero este valor puede ser mayor que 1 o menor de 0, siendo la esperanza matemática de una distribución de Bernouilli, p

INTRODUCCIÓN

También podría ser ε una variable aleatoria de Bernouilli con valores

$$1-(\beta_0+\beta_1 X_{1i}+\beta_2 X_{2i}+...+\beta_k X_{ki})$$

$$y$$

$$(\beta_0+\beta_1 X_{1i}+\beta_2 X_{2i}+...+\beta_k X_{ki})$$

Pero al no ser εi variables normales, la estimación por el método de Mínimos Cuadrados no sería eficiente.

Además si calculamos la varianza de una variable de Bernouilli.

$$V(\varepsilon)=p(1-p)$$

Esta no sería constante. De esta forma no se cumplen las hipótesis de un modelo de regresión lineal.

INTRODUCCIÓN

Desde 1967 se utiliza como si fuera una regresión estándar con datos dicotómicos, esto es, se tiene que Y$\in$ {0,1}

Sea $D_n = \left\{ \mathbf{x_i}, c_i \right\}_{i=1}^{n}$, **, los datos del conjunto de entrenamiento, donde** $\mathbf{x_i} = (x_{1i}, x_{2i}, ..., x_{ki})$ y $c_i = \{0,1\}$, **de forma tal que** c_i**=1, si la observación i-ésima tiene la característica y** c_i**=0 si no la tiene**

La variable dependiente es C con valores 0 y 1 y con probabilidad

$p_i = P(C=1|x_i) = P(C=1|X_1 = x_{1i}, ..., X_k = x_{ki})$,

siendo su media $E(C|x_i) = p_i$ y su Varianza $V(C|x_i) = p_i(1-p_i)$

Buscamos una relación entre la probabilidad de éxito C=1 y las variables predictoras. Los *gráficos* son poco útiles, en general no hay relación entre el valor de la ordenada y los datos de la o las características.

INTRODUCCIÓN: EJEMPLO

En el gráfico del ejemplo no hay relación lineal aceptable que se ajuste a la nube de puntos porque para diferentes valores de la variable SurvRate. Las probabilidades de pertenecer a la clase C=1, están o cercanas a 0 o cercanas a 1

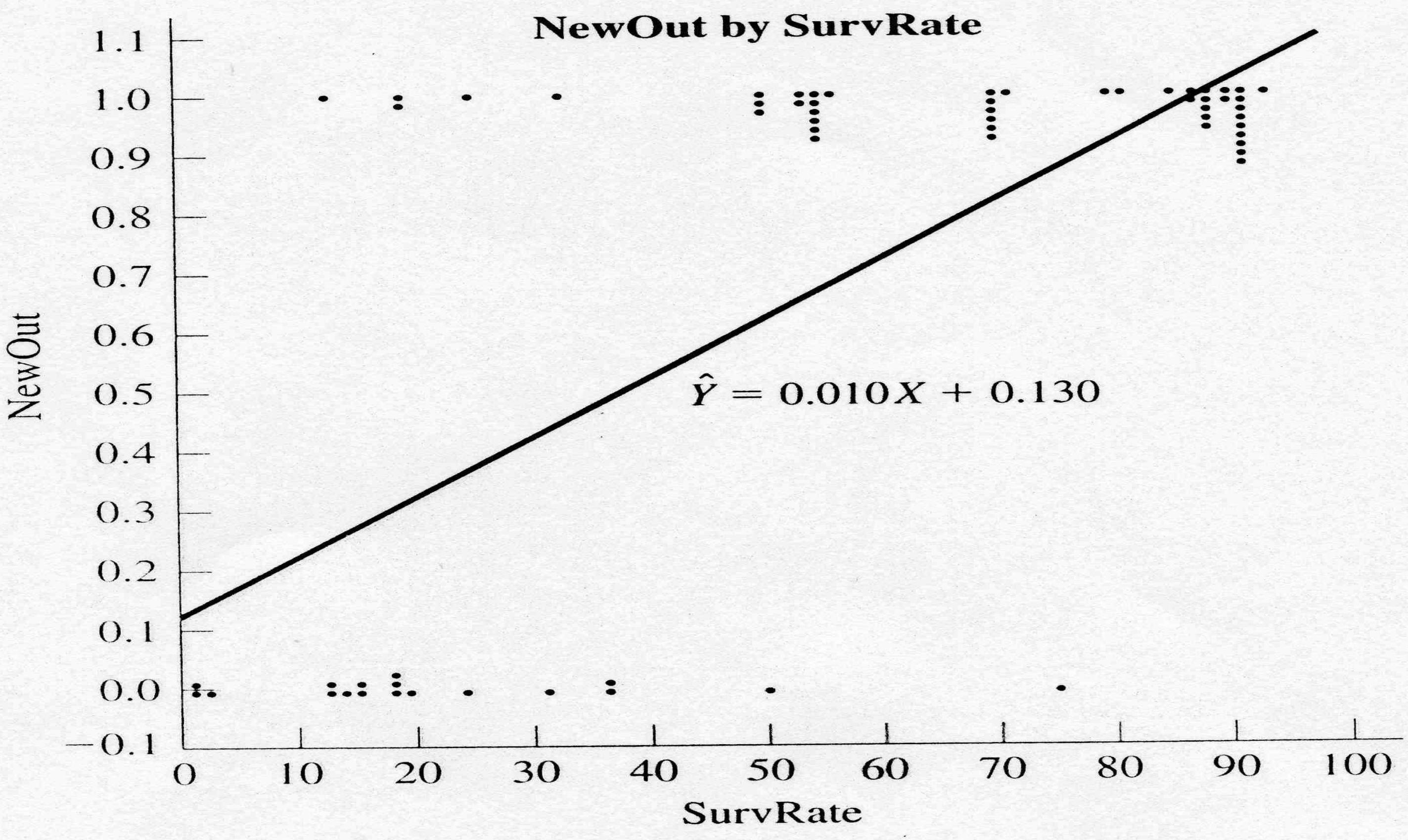


FIGURE 15.7 *Outcome as a function of SurvRate*

INTRODUCCIÓN: EJEMPLO

De esta forma en el gráfico del ejemplo podemos utilizar una función logística, o sigmoide, que se ajuste a la nube de puntos para diferentes valores de la variable SurvRate. Por lo que la ecuación asociada al calculo de la probabilidad de pertenencia a la clase positiva C=1 es la que se muestra en el gráfico.

$$p_i = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_i)}}$$

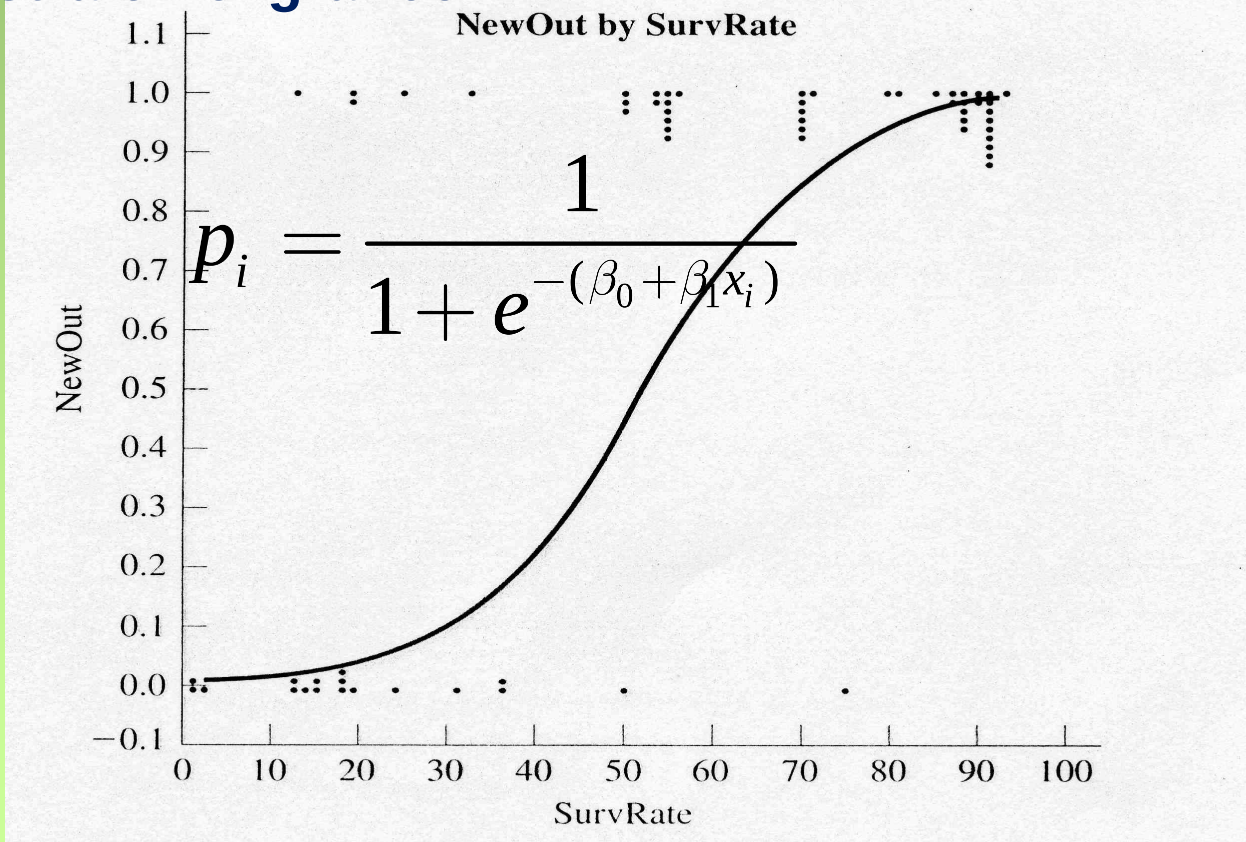


FIGURE 15.8 *More appropriate regression line for predicting outcome*

INTRODUCCIÓN: EJEMPLO

Sea la clase C cuyo valor es igual a 1 cuando el paciente tiene un infarto, y sea x su nivel de colesterol. Para un patrón xi ¿qué relación debería de haber entre Ci y xi?

Si p, la probabilidad de que C=1, debería de ser cercana a 1 para valores de *x* altos, y p=0 para valores de *x* bajos

La relación debería de ser no lineal para muchos valores de *x*: para *x* intermedios casi lineal; y asintótica para valores extremos.

Una función que tiene en cuenta estas hipótesis es la logística, puesto que satisface que pi$\in$[0,1]:

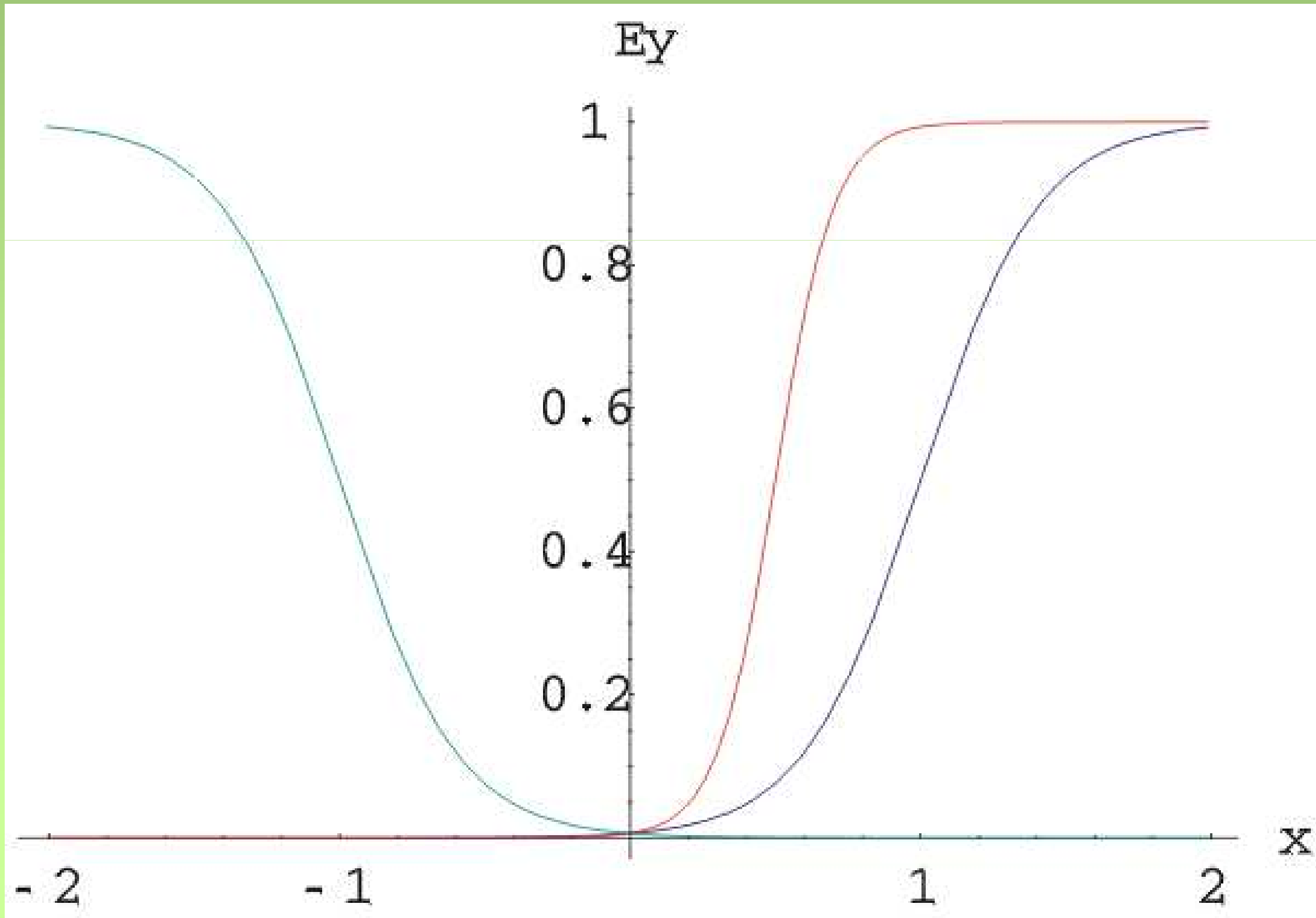
$$p_i = p(C = 1 / x_i) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_i)}} = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}},$$

Curva logística para diferentes valores de β_1

INTRODUCCIÓN: EJEMPLO

Curvas logísticas para diferentes valores de β_1

$$p_i = p(C=1/x_i) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_i)}} = \frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}},$$



INTRODUCCIÓN: EJEMPLO

En general, para garantizar que pi pertenezca al intervalo [0, 1], aplicamos una transformación no lineal:

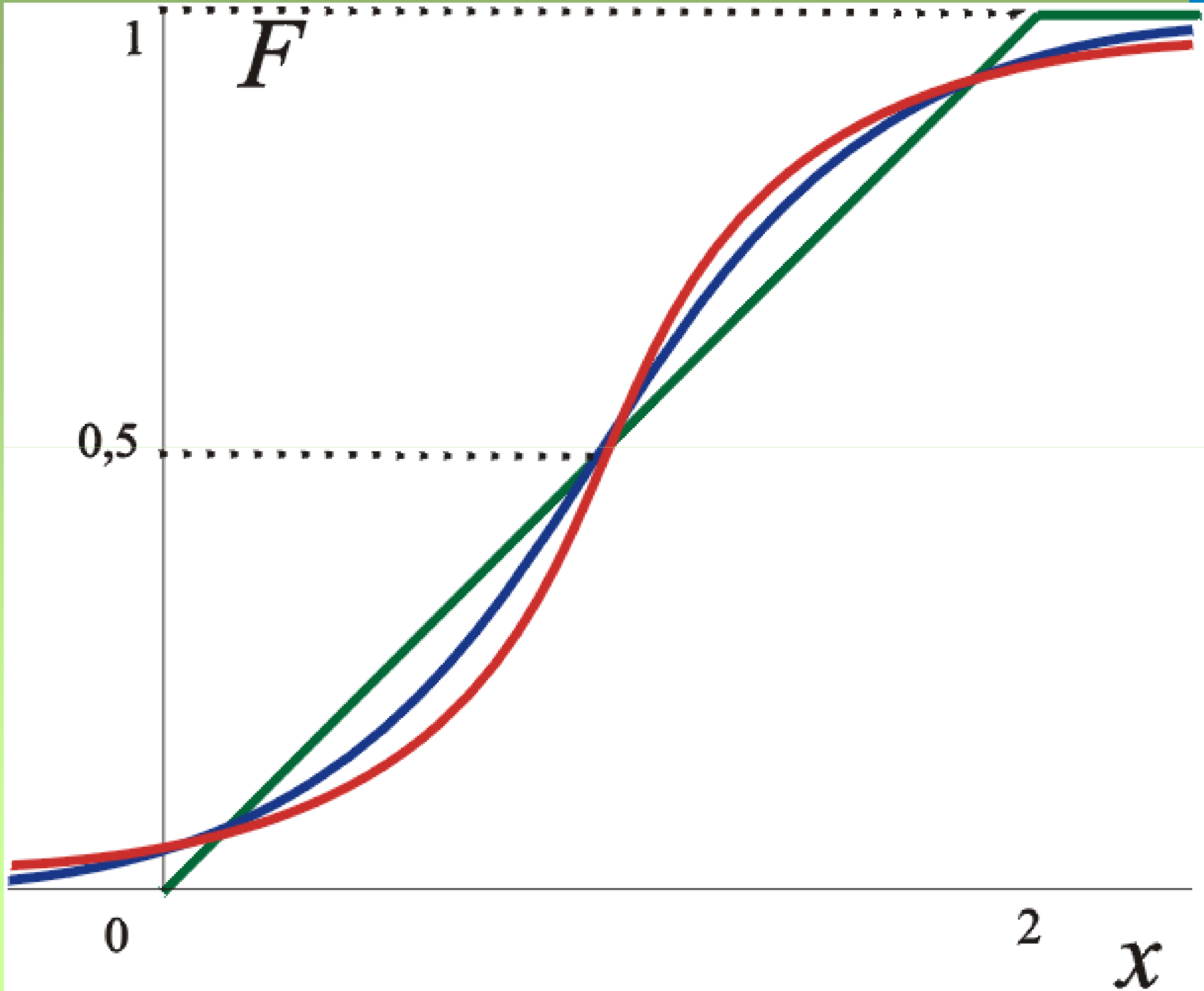
$$p_i = F(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_{\mathbf{i}}),$$

Siendo F cualquier función de distribución de probabilidad

$\boldsymbol{\beta}^{\mathrm{T}} = (\beta_1, ..., \beta_k)$ **es el vector de coeficientes del modelo, y**

$\mathbf{x}_i^T = (x_{1i,} \ldots, x_{ki})$ **es el vector de variables independientes, características o covariables del modelo.**

INTRODUCCIÓN: EJEMPLO



FUNDAMENTO

- **Regresión logística.- Se basa en intentar predecir la probabilidad de pertenencia a cada una de las clases. En primer lugar, suponemos 2 clases, C=0 y C=1 y dos variables de entrada, o covariables X_1 y X_2:**

$$p_{C=0} = 1 - p_{C=1}$$

- **Si consideraramos una Regresión Lineal →**

$$p_{C=1} = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

  - **No tiene sentido → Los valores de p no están acotados entre 0 y 1.**

- **Si consideramos una Regresión Logística →**

$$\ln\left(\frac{P_{C=1}}{1-P_{C=1}}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \quad \rightarrow \quad \boxed{P_{C=1} = \frac{1}{1+e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2)}}}$$

  - **Los valores de p si están acotados entre 0 y 1.**

FUNDAMENTO

- Los parámetros del modelo $\beta_0, \beta_1, \beta_2$ se estiman mediante el método de Máxima Verosimilitud.
- La función –log(verosimilitud) como veremos más adelante es

$$\boxed{\ln(\mathrm{L}) = -\sum_{i=1}^{n} y_i \mathbf{x}_{\mathbf{í}}^{\mathbf{T}} \boldsymbol{\beta} + \sum_{i=1}^{n} \ln(1 + e^{\mathbf{x}_{\mathbf{i}}^{\mathbf{T}} \boldsymbol{\beta}})}$$

  - La estimación de parámetros pasa por maximizar la función -log-verosimilitud (concordancia entre los datos de entrenamiento y los valores de probabilidad del modelo).
  - Si un patrón es de la clase (C=1) → intentaremos que $P_{C=1}$ sea lo más cercana a 1 para ese patrón.

Regresión logística binaria: (1)

- **El modelo de regresión logística es una técnica habitual en estadística en la cual la probabilidad p de pertenencia a la clase positiva de entre dos (caracterizadas por los sucesos C= 1, y C= 0 y asociados a una variable aleatoria de Bernouilli B(p)) está relacionada con un conjunto de valores de las variables explicativas o covariables $\mathbf{x} = (1, x_1, ..., x_k)$ en la forma**

$$\operatorname{logit}(p) = \ln\left(\frac{p}{1-p}\right) = f(\mathbf{x}, \boldsymbol{\beta}) = \beta_0 + \beta_1 x_1 + ...\beta_k x_k = \boldsymbol{\beta}^{\mathbf{T}}\mathbf{x} \tag{1}$$

donde $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, ..., \beta_k)$ **son los coeficientes del modelo a estimar a partir de los datos del conjunto de entrenamiento. A partir de esa expresión, despejando p en (1), se obtiene la probabilidad de éxito como una función no lineal de las covariables.**

$$p = P(C = 1 | \mathbf{x_i}) = P(\mathbf{x}_i; \boldsymbol{\beta}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^{\mathbf{T}}\mathbf{x}_i}} \tag{2}$$

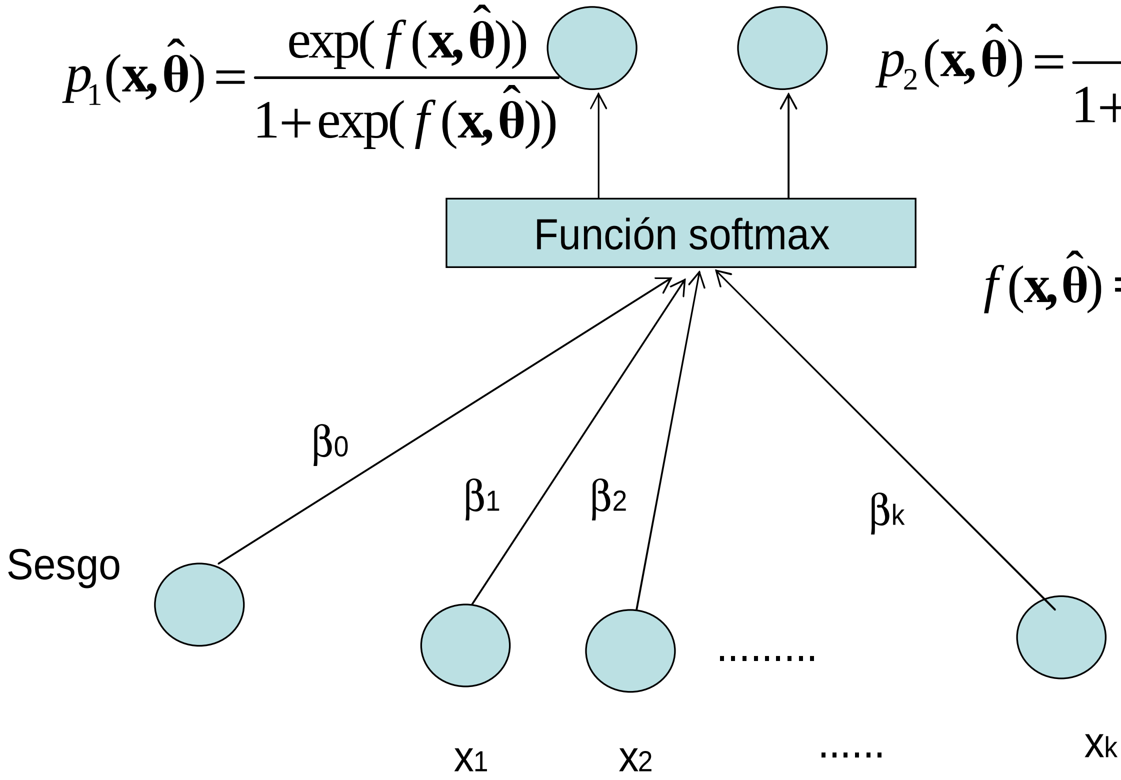
Regresión logística binaria: (2)

La probabilidad de p_2 sobra porque es $1-p_1$

$$p_1(\mathbf{x},\hat{\boldsymbol{\theta}}) = \frac{\exp(f(\mathbf{x},\hat{\boldsymbol{\theta}}))}{1+\exp(f(\mathbf{x},\hat{\boldsymbol{\theta}}))}$$

$$p_2(\mathbf{x},\hat{\boldsymbol{\theta}}) = \frac{1}{1+\exp(f(\mathbf{x},\hat{\boldsymbol{\theta}}))}$$

$$f(\mathbf{x},\hat{\boldsymbol{\theta}}) = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_i x_i$$



Representación de un modelo de Regresión Logística

Regresión logística binaria: Estimación de parámetros

Para dos clases tenemos que

$$\hat{y} = p_i = \hat{p}(C_1 \mid \mathbf{x}) = \frac{1}{1+\exp(-\hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{x})} \text{ es el estimador de } p(C_1 \mid \mathbf{x}),$$

Consideramos que tenemos una muestra de entrenamiento de tamaño n

$$D = \left\{(\mathbf{x_i}, y_i\right\}_{i=1}^{n}, \text{ donde } \mathrm{y_i} = 1 \text{ si } \mathrm{x} \in \mathrm{C_1} \text{ e } \mathrm{y_i} = 0 \text{ si } \mathrm{x} \in \mathrm{C_2}$$

y suponemos ahora que y_i dado x_i sigue una distribución de Bernoulli con probabilidad p **, esto es,** $y_i \mid \mathbf{x_i} \sim B(p_i)$ **, ahora utilizamos el método de máxima verosimilitud para modelar** $p(\mathbf{x} \mid C_1)$

Regresión logística binaria: Estimación de parámetros

La función de verosimilitud para una distribución de Bernouilli es

$$L(\boldsymbol{\beta}, y_{\mathbf{1}}, ..., y_{\mathbf{n}}) = \prod_{i=1}^{n} (p_i)^{y_i} (1 - p_i)^{(1-y_i)}$$

Pero cuando tenemos una función de verosimilitud a maximizar también podemos minimizar una función de error en la forma E=-logL, función a la que denominaremos la Entropía cruzada

$$E = -\ln(L) = -\sum_{i=1}^{n} y_i \ln p_i - \sum_{i=1}^{n} (1 - y_i) \ln(1 - p_i) =$$

$$pero\ como\ \mathrm{p}_i = \frac{1}{1 + e^{-\boldsymbol{\beta}^T \mathbf{x}_i}} = \frac{e^{\boldsymbol{\beta}^T \mathbf{x}_i}}{1 + e^{\boldsymbol{\beta}^T \mathbf{x}_i}}, 1 - p_i = \frac{1}{1 + e^{\boldsymbol{\beta}^T \mathbf{x}_i}}$$

$$= -\sum_{i=1}^{n} y_i \ln(e^{\boldsymbol{\beta}^T \mathbf{x}_i}) + \sum_{i=1}^{n} y_i \ln(1 + e^{\boldsymbol{\beta}^T \mathbf{x}_i}) + \sum_{i=1}^{n} (1 - y_i) \ln(1 + e^{\boldsymbol{\beta}^T \mathbf{x}_i})$$

$$= -\sum_{i=1}^{n} y_i \boldsymbol{\beta}^T \mathbf{x}_i + \sum_{i=1}^{n} \ln(1 + e^{\boldsymbol{\beta}^T \mathbf{x}_i})$$

Así, utilizando la minimización de la función de entropía cruzada, tenemos.

$$E = -\ln(L) = -\sum_{i=1}^{n} y_i \ln p_i - \sum_{i=1}^{n} (1 - y_i) \ln(1 - p_i)$$

$$Sea \ \mathrm{p}_i = \frac{1}{1 + e^{-\beta_0 - \beta_j x_{ij}}}, \ \text{entonces la derivada}$$

$$\frac{\partial p_i}{\partial \beta_j} = p_i (1 - p_i)\, \mathrm{x}_{ij} \quad \text{, y las ecuaciones de cambio son}$$

$$\Delta\beta_j = -\eta \frac{\partial E}{\partial \beta_j} = -\eta \frac{\partial E}{\partial p_i} \frac{\partial p_i}{\partial \beta_j} = \eta \sum_{i=1}^{n} \left(\frac{y_i}{p_i} - \frac{1 - y_i}{1 - p_i} \right) p_i (1 - p_i) x_{ij} =$$

$$= \eta \sum_{i=1}^{n} (y_i - p_i) x_{ij}, \ \text{para j=1,...,k}$$

$$\Delta\beta_0 = -\eta \frac{\partial E}{\partial \beta_0} = \eta \sum_{i=1}^{n} (y_i - p_i)$$

Regresión logística binaria: Estimación de parámetros

ALGORITMO

for j=0,...,k

$\quad \beta_j \leftarrow \mathrm{rand}(-0.01, 0.01)$ inicialización aleatoria

repeat

$\quad$ for j=0,...,k

$\quad\quad \Delta\beta_j \leftarrow 0$

$\quad$ for i $= 1, \ldots, n$

$\quad\quad o \leftarrow 0$

$\quad\quad$ for j=0,...,k

$\quad\quad\quad o \leftarrow o + \beta_j x_{ij}$

$$y \leftarrow sigmoid(o) \equiv \frac{1}{1+e^{-o}}$$

$\quad\quad$ for j=0,...,k

$\quad\quad\quad \Delta\beta_j \leftarrow \Delta\beta_j + (p-y)x_{ij}$

$\quad\quad$ for j=0,...,k

$\quad\quad\quad \beta_j \leftarrow \beta_j + \eta\Delta\beta_j$

until convergence

Regresión logística binaria: Estimación de parámetros

Lo mejor es inicializar los valores de los β_j con valores aleatorios cercanos a 0, por lo general obtenidos a partir de una distribución uniforme U(-0.01, 0.01).

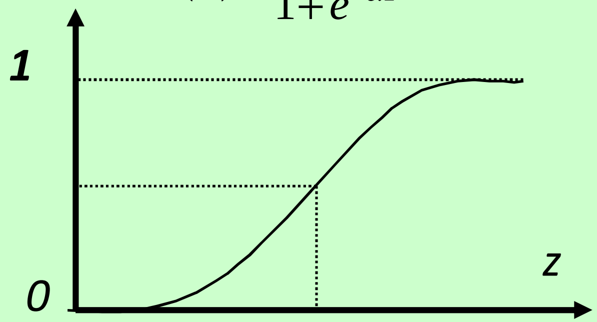
La razón de esto es que si los pesos iniciales son demasiado altos, la suma de estos pesos será también demasiado alta y se podrá saturar la señal de la sigmoide o logística.

Regresión logística binaria: Estimación de parámetros

Podemos ver en la figura adjunta que si los pesos iniciales están próximos a valores intermedios de z, la suma se situará en valores intermedios de la señal, donde la derivada es distinta de cero y puede tener lugar una actualización. Si la suma de pesos es de una gran magnitud o cercana a 0, la derivada de la sigmoide tiende a 0, porque la derivada de la sigmoide es *S'*(*x*) = *S*(*x*)(1–*S*(*x*)) y los pesos no se cambiarán.

Función sigmoide

$$\phi(z) = \frac{1}{1+e^{-\alpha z}}$$



Una vez que se ha completado el entrenamiento o estimación de los parámetros $\boldsymbol{\beta}$, durante la fase de test, calcularemos

$$\hat{p} = \hat{P}(C_1 \mid \mathbf{x}) = \frac{1}{1 + \exp - (\hat{\boldsymbol{\beta}}^{\mathbf{T}} \mathbf{x})}$$

y elegiremos la clase C=1, C1, para el patrón x, si $\hat{p}$ es mayor de 0.5; en otro caso elegiremos la clase C=0, C2

Posibilidad y cociente de posibilidades

- **Consideremos el conjunto de datos**

		Delincuente		
		Si	**No**	**Total**
Tetosterona	**Normal**	402	3614	4016
	Alta	101	345	446
		503	3959	4462

- **Las posibilidades de ser delincuente si se pertenece al grupo Normal son (frecuencia de delincuente)/(frecuencia de no delincuente)**

$p_{\text{delincuente}}/p_{\text{no delincuente}} = p_{\text{delincuente}}/1–p_{\text{delincuente}}$

$p_{\text{delincuente}} = 402/4016 = 0.1001$

$p_{\text{no delincuente}} = 1 – 0.1001 = 0.8889$

$odds = p/(1–p)$

posibilidad u *odds* (delincuente) = 0.1001/0.8999 = 0.111

Posibilidad y cociente de posibilidades

La posibilidad de ser no delincuente en el grupo Normal es la reciproca: 0.8999/0.1001=8.99

Para el grupo de tetosterona Alta

posibilidad(delincuente) = 101/345 = 0.293

posibilidad(no delincuente) = 345/101 = 3.416

Para el grupo de tetosterona Normal

posibilidad (delincuente) = 0.1001/0.8999 = 0.111

Cuando vamos desde el grupo Normal a Alta, las posibilidades de ser delincuente se acercan al triple: 0.293/0.111 = 2.64

Cociente de posibilidades: Una persona es 2.64 veces más probable que sea delincuente con altos niveles de tetosterona que con normales.

Ejemplo Pima con SPSS

Preparar los datos: Variable clase, Variables independientes

- Variable de selección → Entrenamiento / Generalización

*ejemploADLpimadef.sav [Conjunto_de_datos1] - Editor de datos SPSS

Archivo Edición Ver Datos Transformar Analizar Gráficos Utilidades Ventana ?

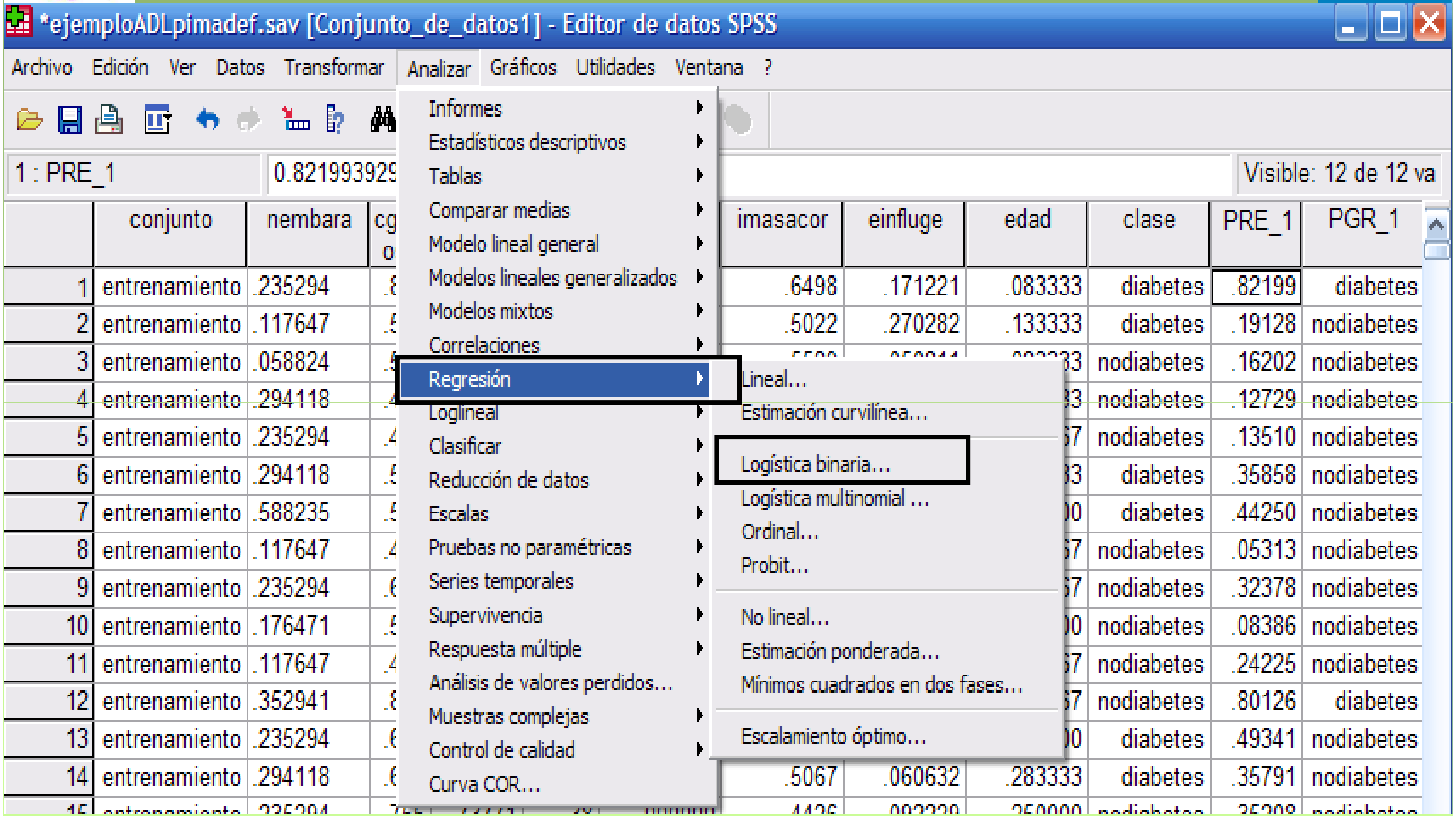
1 : PRE_1 0.821993929850541 Visible: 12 de 12 va

	conjunto	nembara	cglucosa	pdiastol	gdobpiel	ninsulin	imasacor	einfluge	edad	clase	PRE_1	PGR_1
1	entrenamiento	.235294	.855	.59016	.00	.000000	.6498	.171221	.083333	diabetes	.82199	diabetes
2	entrenamiento	.117647	.525	.65574	.45	.225768	.5022	.270282	.133333	diabetes	.19128	nodiabetes
3	entrenamiento	.058824	.530	.62295	.00	.000000	.5589	.050811	.083333	nodiabetes	.16202	nodiabetes
4	entrenamiento	.294118	.495	.60656	.27	.000000	.4322	.053373	.183333	nodiabetes	.12729	nodiabetes
5	entrenamiento	.235294	.455	.57377	.32	.104019	.4933	.157131	.016667	nodiabetes	.13510	nodiabetes
6	entrenamiento	.294118	.580	.60656	.29	.000000	.4814	.248506	.233333	diabetes	.35858	nodiabetes
7	entrenamiento	.588235	.540	.54098	.00	.000000	.4829	.082835	.350000	diabetes	.44250	nodiabetes
8	entrenamiento	.117647	.440	.60656	.19	.062648	.4322	.064475	.016667	nodiabetes	.05313	nodiabetes
9	entrenamiento	.235294	.600	.55738	.00	.000000	.4411	.269428	.216667	nodiabetes	.32378	nodiabetes
10	entrenamiento	.176471	.530	.59016	.00	.000000	.3845	.055081	.100000	nodiabetes	.08386	nodiabetes
11	entrenamiento	.117647	.410	.42623	.22	.135934	.4247	.692143	.066667	nodiabetes	.24225	nodiabetes
12	entrenamiento	.352941	.825	.55738	.26	.198582	.5007	.236123	.466667	nodiabetes	.80126	diabetes
13	entrenamiento	.235294	.625	.57377	.18	.144208	.4307	.455167	.400000	diabetes	.49341	nodiabetes
14	entrenamiento	.294118	.620	.60656	.00	.000000	.5067	.060632	.283333	diabetes	.35791	nodiabetes
15	entrenamiento	.235294	.755	.73771	.38	.000000	.4426	.092229	.250000	nodiabetes	.35208	nodiabetes
16	entrenamiento	.117647	.415	.53279	.28	.078014	.5484	.235269	.050000	nodiabetes	.16005	nodiabetes
17	entrenamiento	.000000	.655	.00000	.00	.000000	.6438	.081981	.083333	diabetes	.70419	diabetes
18	entrenamiento	.176471	.540	.50820	.24	.000000	.3875	.061913	.066667	nodiabetes	.10256	nodiabetes
19	entrenamiento	.000000	.900	.63934	.63	.016549	.8852	1.000000	.066667	diabetes	.99062	diabetes
20	entrenamiento	.117647	.460	.62295	.20	.000000	.3607	.691716	.116667	nodiabetes	.17368	nodiabetes

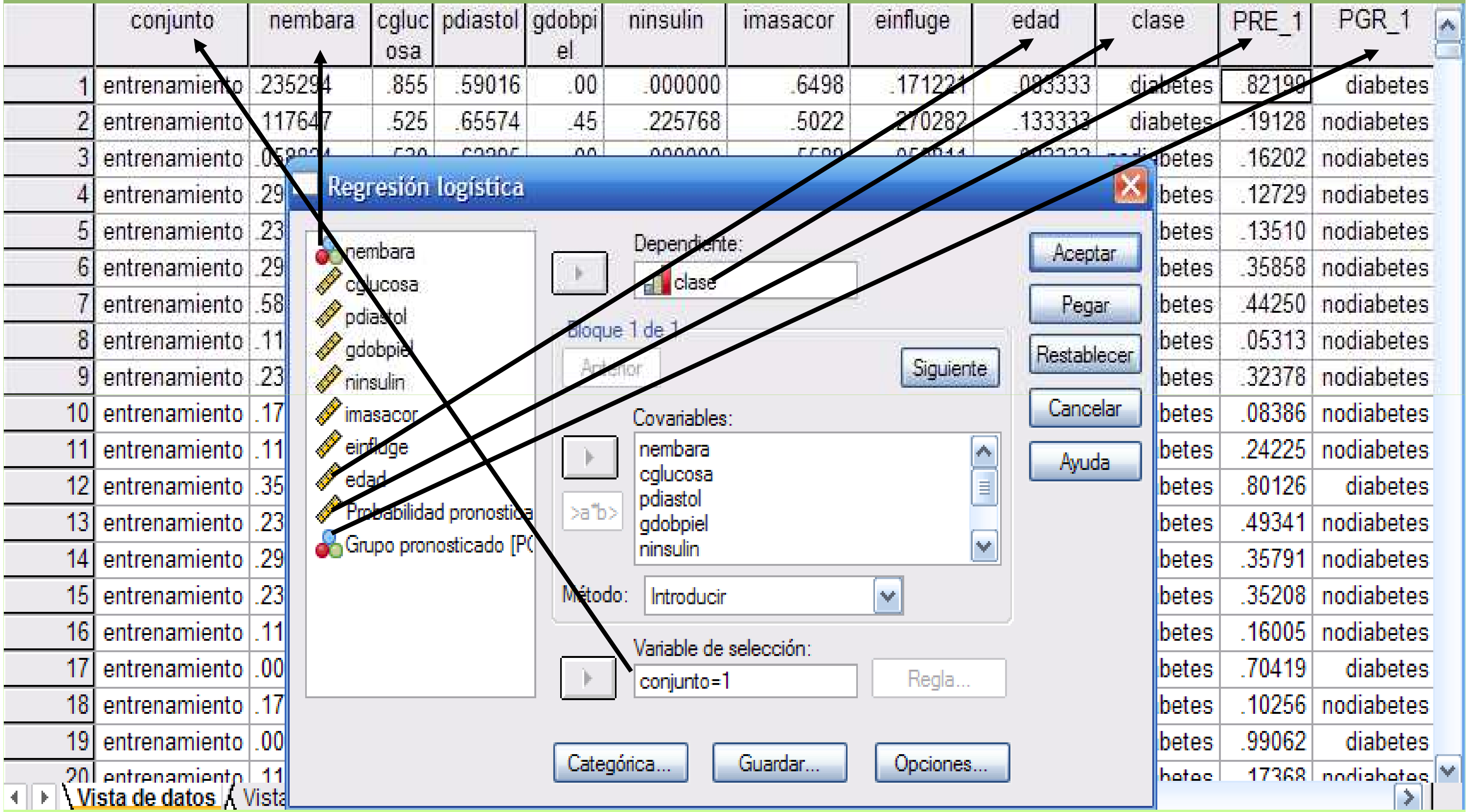
Vista de datos / Vista de variables

Área de recuento de casos SPSS El procesador está preparado

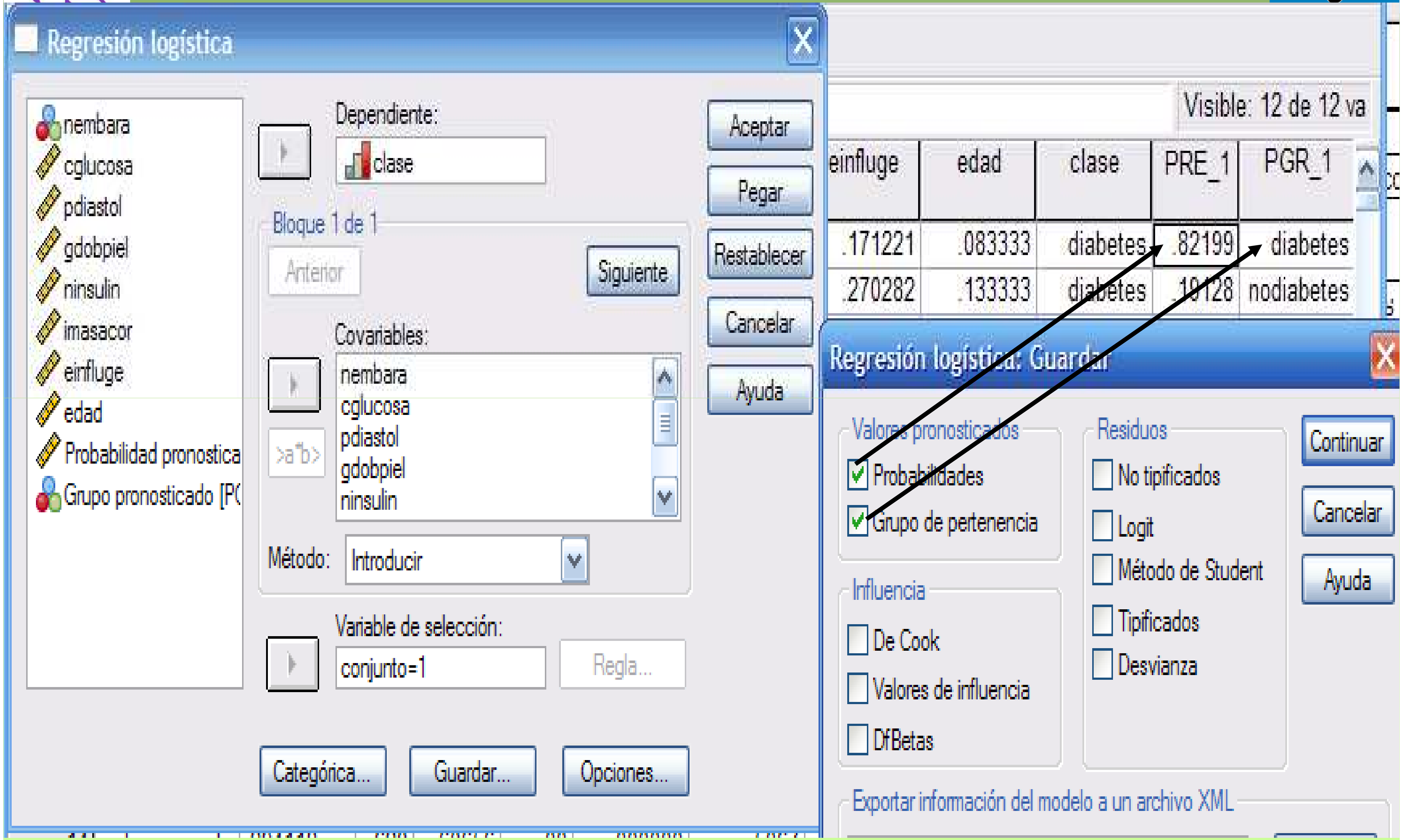
Regresión logística binaria: SPSS



Regresión logística binaria: SPSS



Regresión logística binaria: SPSS



Salida de SPSS de la Regresión Logística Binaria

Resumen del procesamiento de los casos

Casos no ponderados [a]		N	Porcentaje
Casos seleccionados	Incluidos en el análisis	575	74.9
	Casos perdidos	0	.0
	Total	575	74.9
Casos no seleccionados		193	25.1
Total		768	100.0

a. Si está activada la ponderación, consulte la tabla de clasificación para ver el número total de casos.

Codificación de la variable dependiente

Valor original	Valor interno
nodiabetes	0
diabetes	1

Pruebas omnibus sobre los coeficientes del modelo

		Chi-cuadrado	gl	Sig.
Paso 1	Paso	188.997	8	.000
	Bloque	188.997	8	.000
	Modelo	188.997	8	.000

Tabla de clasificación[c]

Observado			Pronosticado: Casos seleccionados [a]: clase: nodiabetes	clase: diabetes	Porcentaje correcto	Casos no seleccionados [b]: clase: nodiabetes	clase: diabetes	Porcentaje correcto
Paso 1	clase	nodiabetes	324	47	87.3	114	15	88.4
		diabetes	85	119	58.3	25	39	60.9
	Porcentaje global				77.0			79.3

a. Casos seleccionados conjunto EQ 1

b. Casos no seleccionados conjunto NE 1

c. El valor de corte es .500

Salida de SPSS

Resumen de los modelos

Paso	-2 log de la verosimilitud	R cuadrado de Cox y Snell	R cuadrado de Nagelkerke
1	558.914[a]	.280	.385

a. La estimación ha finalizado en el número de iteración 5 porque las estimaciones de los parámetros han cambiado en menos de .001.

Variables en la ecuación

		B	E.T.	Wald	gl	Sig.	Exp(B)
Paso 1[a]	nembara	2.260	.637	12.588	1	.000	9.581
	cglucosa	5.627	.815	47.656	1	.000	277.734
	pdiastol	-2.118	.783	7.309	1	.007	.120
	gdobpiel	.024	.785	.001	1	.976	1.024
	ninsulin	-.399	.878	.207	1	.649	.671
	imasacor	6.413	1.145	31.382	1	.000	609.534
	einfluge	2.436	.773	9.940	1	.002	11.429
	edad	1.483	.678	4.786	1	.029	4.405
	Constante	-7.270	.768	89.537	1	.000	.001

a. Variable(s) introducida(s) en el paso 1: nembara, cglucosa, pdiastol, gdobpiel, ninsulin, imasacor, einfluge, edad.

Logistic Weka. Multiclase

Logistic es una implementación alternativa para construir y utilizar un modelo de regresión logística multiclase o multinomial con un estimador estricto para prevenirnos frente al sobreentrenamiento penalizando modelos con coeficientes grandes, basado en un trabajo de Cessie and van Houwelingen (1992).

Las Figuras siguientes muestran su salida sobre la base de datos iris.

Los coeficientes de las funciones de regresión logística se muestran en forma de tabla, uno para cada valor de la clase a excepción de la última clase

Logistic Weka. Multiclase

Dados k atributos de entrada y J clases, la probabilidad de predecir la pertenencia a la clase j dado un patrón x viene dada por

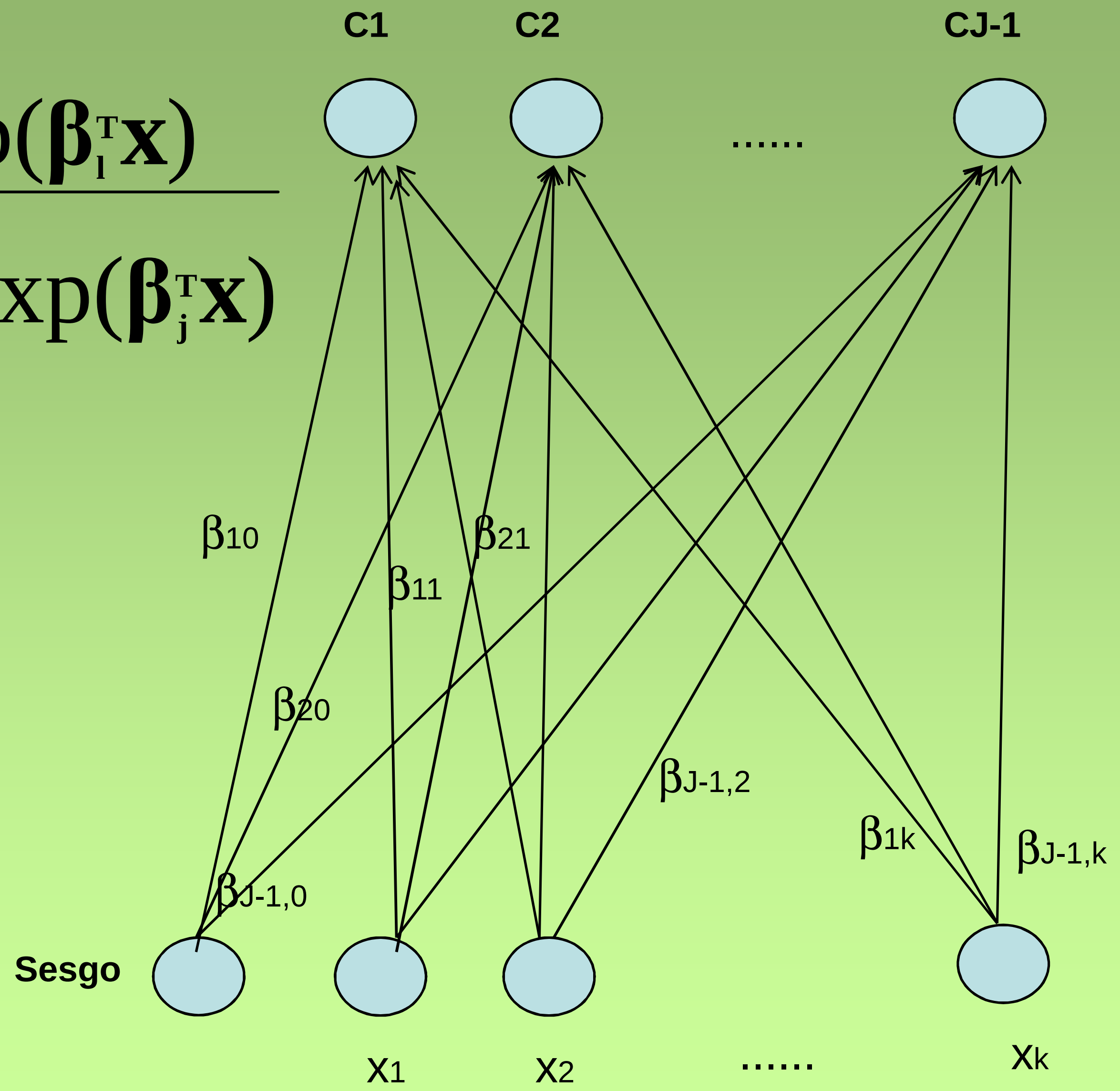
$$p(\mathbf{x} \in C_j) = \frac{\exp(\beta_0^j + \sum_{i=1}^{k} \beta_i^j x_i)}{1 + \sum_{j=1}^{J-1} \exp(\beta_0^j + \sum_{i=1}^{k} \beta_i^j x_i)} \quad \text{para j=1,...,J-1}$$

$$\mathrm{p}(\mathbf{x} \in \mathrm{C}_J) = 1 - \sum_{j=1}^{J-1} p(\mathbf{x} \in C_j),$$

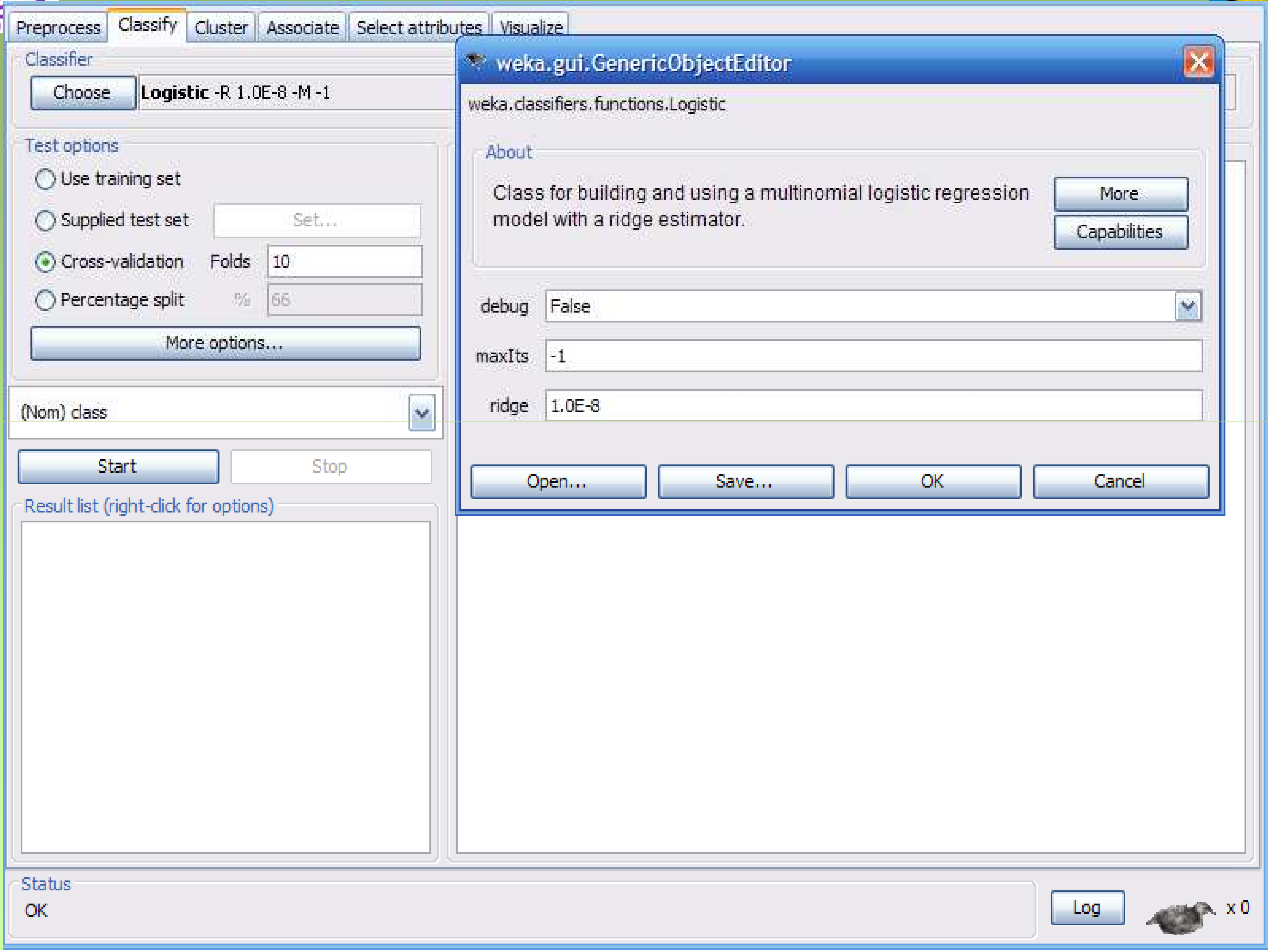
pues la suma de las probabilidades es1

Grafo de una regresión logística multiclase

$$P(C=C1)=\frac{\exp(\boldsymbol{\beta}_1^T\mathbf{x})}{1+\sum_{j=1}^{J-1}\exp(\boldsymbol{\beta}_j^T\mathbf{x})}$$



Logistic Weka. Multiclase



Logistic Weka. Multiclase

```
There are some modifications, however, compared to the paper of leCessie and van Houwelingen(1992):

If there are k classes for n instances with m attributes, the parameter matrix B to be calculated will be an m*(k-1)
matrix.
```

En este caso hay k clases, n patrones y m atributos o variables independientes

```
The probability for class j with the exception of the last class is

Pj(Xi) = exp(XiBj)/((sum[j=1..(k-1)]exp(Xi*Bj))+1)

The last class has probability

1-(sum[j=1..(k-1)]Pj(Xi))
        = 1/((sum[j=1..(k-1)]exp(Xi*Bj))+1)

The (negative) multinomial log-likelihood is thus:

L = -sum[i=1..n]{
        sum[j=1..(k-1)](Yij * ln(Pj(Xi)))
        +(1 - (sum[j=1..(k-1)]Yij))
        * ln(1 - sum[j=1..(k-1)]Pj(Xi))
        } + ridge * (B^2)
```

$$p(\mathbf{x} \in Cj) = \frac{\exp(\beta_0^j + \sum_{i=1}^{k} \beta_i^j x_i)}{1 + \sum_{j=1}^{J-1} \exp(\beta_0^j + \sum_{i=1}^{k} \beta_i^j x_i)} \quad \text{para j=1,...,J-1}$$

$$\mathrm{p}(\mathbf{x} \in \mathrm{C}_J) = 1 - \sum_{j=1}^{J-1} p(\mathbf{x} \in C_j)$$

para dos clases

$$E = -\ln(L) = -\sum_{i=1}^{n} y_i \ln p_i - \sum_{i=1}^{n} (1 - y_i) \ln(1 - p_i) =$$

$$= -\sum_{i=1}^{n} y_i \boldsymbol{\beta}^T \mathbf{x}_i + \sum_{i=1}^{n} \ln(1 + e^{\boldsymbol{\beta}^T \mathbf{x}_i}), \text{ siendo } \mathrm{p}_i = \frac{1}{1 + e^{-\boldsymbol{\beta}^T \mathbf{x}_i}} = \frac{e^{\boldsymbol{\beta}^T \mathbf{x}_i}}{1 + e^{\boldsymbol{\beta}^T \mathbf{x}_i}}$$

```
In order to find the matrix B for which L is minimised, a Quasi-Newton Method is used to search for the optimized values
of the m*(k-1) variables.  Note that before we use the optimization procedure, we 'squeeze' the matrix B into a m*(k-1)
vector.  For details of the optimization procedure, please check weka.core.Optimization class.

Although original Logistic Regression does not deal with instance weights, we modify the algorithm a little bit to handle
the instance weights.

For more information see:
```


```
le Cessie, S., van Houwelingen, J.C. (1992). Ridge Estimators in Logistic Regression. Applied Statistics. 41(1):191-201.
```

Regresión logística binaria, Indias Pima: Estimación de parámetros

Logistic -R 1.0E-8 -M -1

Coeficientes del modelos

preg	-0.1232
plas	-0.0352
pres	0.0133
skin	-0.0006
insu	0.0012
mass	-0.0897
pedi	-0.9452
age	-0.0149
Intercept	8.4047

Odds Ratios...

Variable	Class tested_negative
preg	0.8841
plas	0.9654
pres	1.0134
skin	0.9994
insu	1.0012
mass	0.9142
pedi	0.3886
age	0.9852

Ecuación del modelo de regresión logística

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \ldots \beta_k x_k =$$

$$= 8,4047 - 0,1232 \times preg$$

$$-0,0352 \times plas + 0.0133 \times pres$$

$$-0,0006 \times skin + 0,0012 \times insu$$

$$-0,0897 \times mass - 0,9452 \times pedi$$

$$-0,0149 \times age$$

Regresión logística binaria, Pima: Estimación de parámetros

Preprocess | Classify

Classifier

Choose **Logistic** -R 1.0E-8 -M -1

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 75

More options...

(Nom) class

Start | Stop

Result list (right-click for options)

11:00:07 - functions.Logistic

Classifier output

preg	-0.1232
plas	-0.0352
pres	0.0133
skin	-0.0006
insu	0.0012
mass	-0.0897
pedi	-0.9452
age	-0.0149
Intercept	8.4047

Odds Ratios...

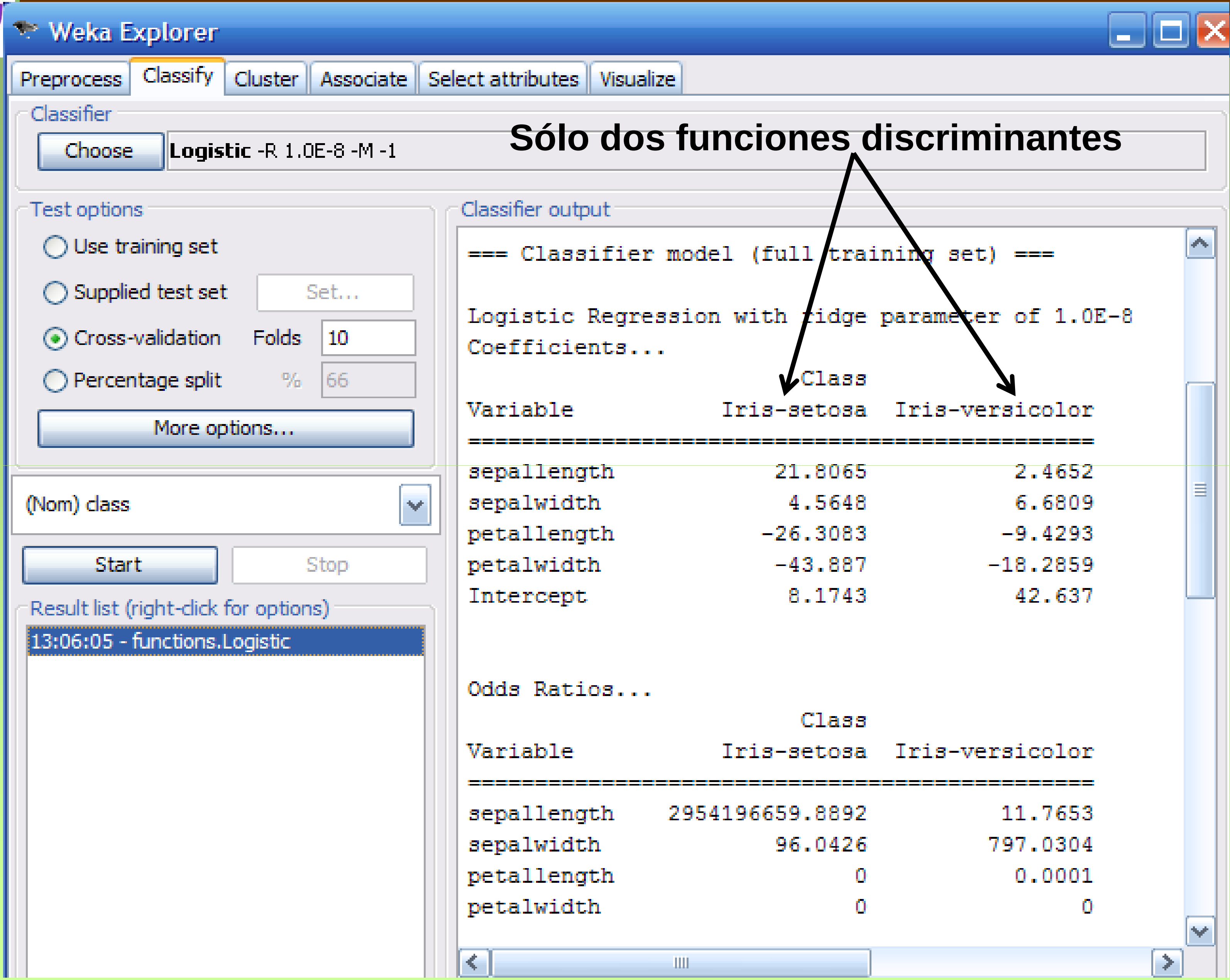
Variable	Class tested_negative
preg	0.8841
plas	0.9654
pres	1.0134
skin	0.9994
insu	1.0012
mass	0.9142
pedi	0.3886
age	0.9852

Los odds que están por encima de 1 indican que las variables asociadas influyen positivamente sobre la probabilidad de que el patrón pertenezca a la clase Y=1

$$P_{Y=1} = \frac{1}{1+e^{-\beta_1 X_1}} = \frac{1}{1+e^{0,1232}} = 0,4692$$

$$\text{Odd Preg} = \frac{0,4692}{0,5307} = 0,884$$

Logistic Weka. Multiclase, Iris, entrenamiento



Logistic Weka. Multiclase, Iris, entrenamiento

Cálculos de probabilidades de pertenencia a las clases

Regla de decisión, el patrón pertenece a la clase con mayor probabilidad

$$p(\mathbf{x} \in C_{setosa}) = \frac{\exp(\beta_0^{setosa} + \sum_{i=1}^{k} \beta_i^{setosa} x_i)}{1 + \exp(\beta_0^{setosa} + \sum_{i=1}^{k} \beta_i^{setosa} x_i) + \exp(\beta_0^{ver} + \sum_{i=1}^{k} \beta_i^{ver} x_i)} =$$

$$\frac{e^{8,1743 + 21,8065 \times sele + 4,5648 \times sewi - 26,3083 \times pele - 43,887 \times pewi}}{1 + e^{8,1743 + 21,8065 \times sele + 4,5648 \times sewi - 26,3083 \times pele - 43,887 \times pewi} + e^{42,637 + 2,4652 \times sele + 6,6809 \times sewi - 9,4293 \times pele - 18,2859 \times pewi}}$$

$$p(\mathbf{x} \in C_{versicolor}) = \frac{\exp(\beta_0^{versicolor} + \sum_{i=1}^{k} \beta_i^{versicolor} x_i)}{1 + \exp(\beta_0^{setosa} + \sum_{i=1}^{k} \beta_i^{setosa} x_i) + \exp(\beta_0^{ver} + \sum_{i=1}^{k} \beta_i^{ver} x_i)} =$$

$$\frac{e^{42,637 + 2,4652 \times sele + 6,6809 \times sewi - 9,4293 \times pele - 18,2859 \times pewi}}{1 + e^{8,1743 + 21,8065 \times sele + 4,5648 \times sewi - 26,3083 \times pele - 43,887 \times pewi} + e^{42,637 + 2,4652 \times sele + 6,6809 \times sewi - 9,4293 \times pele - 18,2859 \times pewi}}$$

$$\mathrm{p}(\mathbf{x} \in \mathrm{C}_{virgínica}) = 1 - p(\mathbf{x} \in C_{setosa}) - \mathrm{p}(\mathbf{x} \in \mathrm{C}_{versicolor})$$

Logistic Weka. Multiclase, toda la base de datos Iris

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)

13:06:05 - functions.Logistic

Classifier output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.0287
Root mean squared error                  0.1424
Relative absolute error                  6.456  %
Root relative squared error             30.2139 %
Coverage of cases (0.95 level)          98.6667 %
Mean rel. region size (0.95 level)      35.5556 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Iris-setosa
                 0.92      0.02       0.958     0.92      0.939      0.97     Iris-versicolor
                 0.96      0.04       0.923     0.96      0.941      0.975    Iris-virginica
Weighted Avg.    0.96      0.02       0.96      0.96      0.96       0.982

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 46  4 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

Metodología 1 frente a todos

Setosa frente a todos

Versicolor frente a todos

Virgínica frente a todos

Media ponderada

Status

OK

Log

x 0

Logistic Weka. Iris sólo entrenamiento

Scheme: weka.classifiers.functions.Logistic -R 1.0E-8 -M -1
Relation: train_iris
Instances: 111
Attributes: 5
x0
x1
x2
x3
Class

$$p(\mathbf{x} \in Cj) = \frac{\exp(\beta_0^j + \sum_{i=1}^{k} \beta_i^j x_i)}{1 + \sum_{j=1}^{J-1} \exp(\beta_0^j + \sum_{i=1}^{k} \beta_i^j x_i)} \quad \text{para j=1,...,J-1}$$

$$\mathrm{p}(\mathbf{x} \in \mathrm{C}_J) = 1 - \sum_{j=1}^{J-1} p(\mathbf{x} \in C_j)$$

Test mode: evaluate on training data
=== Classifier model (full training set) ===
Logistic Regression with ridge parameter of 1.0E-8 (parámetro de reborde o de polarización constante)
Coefficients...

	Class	
Variable	y0	y1
x0	158.9514	68.7127
x1	82.8885	126.0823
x2	-326.5405	-261.0708
x3	-678.7102	-639.4349
Intercept	1456.6609	1605.6171

Odds Ratios...

	Class	
Variable	y0	y1
x0	1.0757170472577295E69	6.943330678752426E29
x1	9.954769004537487E35	5.713029198452447E54
x2	0	0
x3	0	0

Logistic Weka. Iris sólo entrenamiento

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	111	100 %
Incorrectly Classified Instances	0	0 %
Kappa statistic		1
Mean absolute error		0.0005
Root mean squared error		0.0029
Relative absolute error		0.1025 %
Root relative squared error		0.6133 %
Coverage of cases (0.95 level)		100 %
Mean rel. region size (0.95 level)		33.3333 %
Total Number of Instances		111

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	y0
	1	0	1	1	1	1	y1
	1	0	1	1	1	1	y2
Wei Avg.	1	0	1	1	1	1	

=== Confusion Matrix ===

```
  a  b  c   <-- classified as
 37  0  0 |  a = y0
  0 37  0 |  b = y1
  0  0 37 |  c = y2
```

Logistic Weka. Iris testing

Scheme: weka.classifiers.functions.Logistic -R 1.0E-8 -M -1
Relation: train_iris
Instances: 111
Attributes: 5
x0
x1
x2
x3
Class
Test mode: user supplied test set: size unknown (reading incrementally)
=== Classifier model (full training set) ===
Logistic Regression with ridge parameter of 1.0E-8
Coefficients...

	Class	
Variable	y0	y1
x0	158.9514	68.7127
x1	82.8885	126.0823
x2	-326.5405	-261.0708
x3	-678.7102	-639.4349
Intercept	1456.6609	1605.6171

Odds Ratios...

	Class	
Variable	y0	y1
x0	1.0757170472577295E69	6.943330678752426E29
x1	9.954769004537487E35	5.713029198452447E54
x2	0	0
x3	0	0

Logistic Weka. Iris testing

=== Evaluation on test set ===
=== Summary ===

37/39=0.948

Correctly Classified Instances	37	94.8718 %
Incorrectly Classified Instances	2	5.1282 %
Kappa statistic		0.9231
Mean absolute error		0.0342
Root mean squared error		0.1849
Relative absolute error		7.6957 %
Root relative squared error		39.2232 %
Coverage of cases (0.95 level)		94.8718 %
Mean rel. region size (0.95 level)		33.3333 %
Total Number of Instances		39

$$\text{Recall=Sensibilidad}=\frac{\text{TP}}{\text{TP+FN}}$$

$$\text{Precision}=\frac{\text{TP}}{\text{TP+FP}}$$

$$F=\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision+Recall}}$$

$$\text{Kappa}=\frac{\text{TP+TN-E(TP+TN)}}{\text{TP+TN+FP+FN-E(TP+TN)}}$$

=== Detailed Accuracy By Class ===

Metodología 1 frente a todos

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	y0
	1	0.077	0.867	1	0.929	0.975	y1
	0.846	0	1	0.846	0.917	1	y2
Wei Avg.	0.949	0.026	0.956	0.949	0.948	0.992	

=== Confusion Matrix ===

```
  a  b  c   <-- classified as
 13  0  0 |  a = y0
  0 13  0 |  b = y1
  0  2 11 |  c = y2
```

11/13=0.846

13/15=0.8666

$$\begin{pmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{pmatrix}$$

$$\begin{pmatrix} TP & FN \\ FP & TN \end{pmatrix}$$

$$\text{Precision}=\frac{TP}{TP+FP} \quad \text{Recall}=\text{Sensibilidad}=\frac{TP}{TP+FN} \quad F=\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$$

	TP Rate	FP Rate	Precision	Recall	F-Measure	Area AUC	Class
	1	0	1	1	1	1	y0
	1	2/26=0.077	13/15=0.867	1	0.929	0.975	y1
	0.846	0	1	11/13=0.846	0.917	1	y2
Wei Avg.	0.949	0.026	0.956	0.949	0.948	0.992	

=== Confusion Matrix ===

a	b	c	<-- classified as,1 vs All	Class 1		Class 2		Class 3	
13	0	0	a = y0	13	0	13	0	11	2
0	13	0	b = y1	0	26	2	24	0	26
0	2	11	c = y2						

SimpleLogistic Weka. Multiclase

Este método de regresión logística utiliza modelos fijándolos mediante el procedimiento *LogitBoost* mediante funciones sencillas de regresión como métodos de aprendizaje de base y determinando cuantas iteraciones tenemos que realizar utilizando validación cruzada, considerando además selección automática de atributos, Landwehr et al., 2005. *SimpleLogistic* genera un árbol logístico degenerado compuesto por un solo nodo, y soporta las opciones que se aplican a Logistic Model Tree, LMT.

SimpleLogistic Weka. Multiclase

Por debajo de la tabla de los coeficientes de regresión hay una segunda tabla que da una estimación de los cocientes de las probabilidades para cada atributo de entrada y cada clase. Para un atributo dado, este valor da una indicación de su influencia en la clase cuando los valores de los otros atributos se mantienen fijos

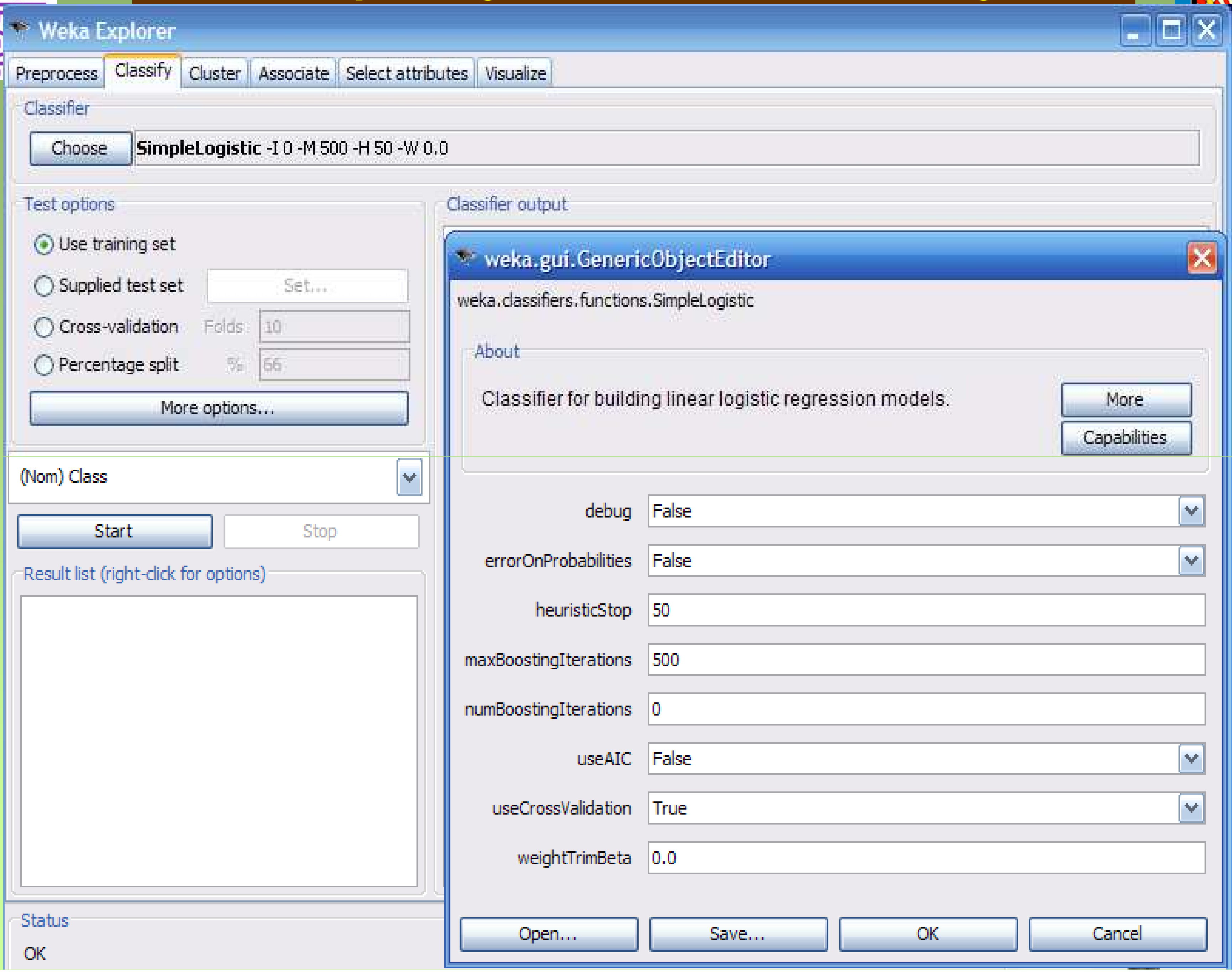
SimpleLogistic Weka.

***LogicBoost*, realiza una regresión logística aditiva, de la misma forma que *AdaBoostM1*, el algoritmo se puede acelerar especificando un umbral de poda de pesos. El número más apropiado de iteraciones se puede determinar utilizando un procedimiento interno de validación cruzada sobre el conjunto de entrenamiento (casi siempre es un 5-fold)**

SimpleLogistic Weka.

Existe un parámetro de contracción que puede acoplarse para prevenir el sobreentrenamiento, también se puede elegir remuestreo en vez de reponderación. Se puede realizar remuestreo si el clasificador base no puede utilizar instancias ponderadas (aunque se puede forzar un remuestreo en cualquier caso)

SimpleLogistic Weka. Iris training



SimpleLogistic Weka. Iris training

weka.gui.GenericObjec

weka.classifiers.functions.Simple

About

Classifier for building line

debug	False
errorOnProbabilities	False
heuristicStop	50
maxBoostingIterations	500
numBoostingIterations	0
useAIC	False
useCrossValidation	True
weightTrimBeta	0.0

Open...

Status

OK

Information

errorOnProbabilities -- Use error on the probabilties as error measure when determining the best number of LogitBoost iterations. If set, the number of LogitBoost iterations is chosen that minimizes the root mean squared error (either on the training set or in the cross-validation, depending on useCrossValidation).

heuristicStop -- If heuristicStop > 0, the heuristic for greedy stopping while cross-validating the number of LogitBoost iterations is enabled. This means LogitBoost is stopped if no new error minimum has been reached in the last heuristicStop iterations. It is recommended to use this heuristic, it gives a large speed-up especially on small datasets. The default value is 50.

maxBoostingIterations -- Sets the maximum number of iterations for LogitBoost. Default value is 500, for very small/large datasets a lower/higher value might be preferable.

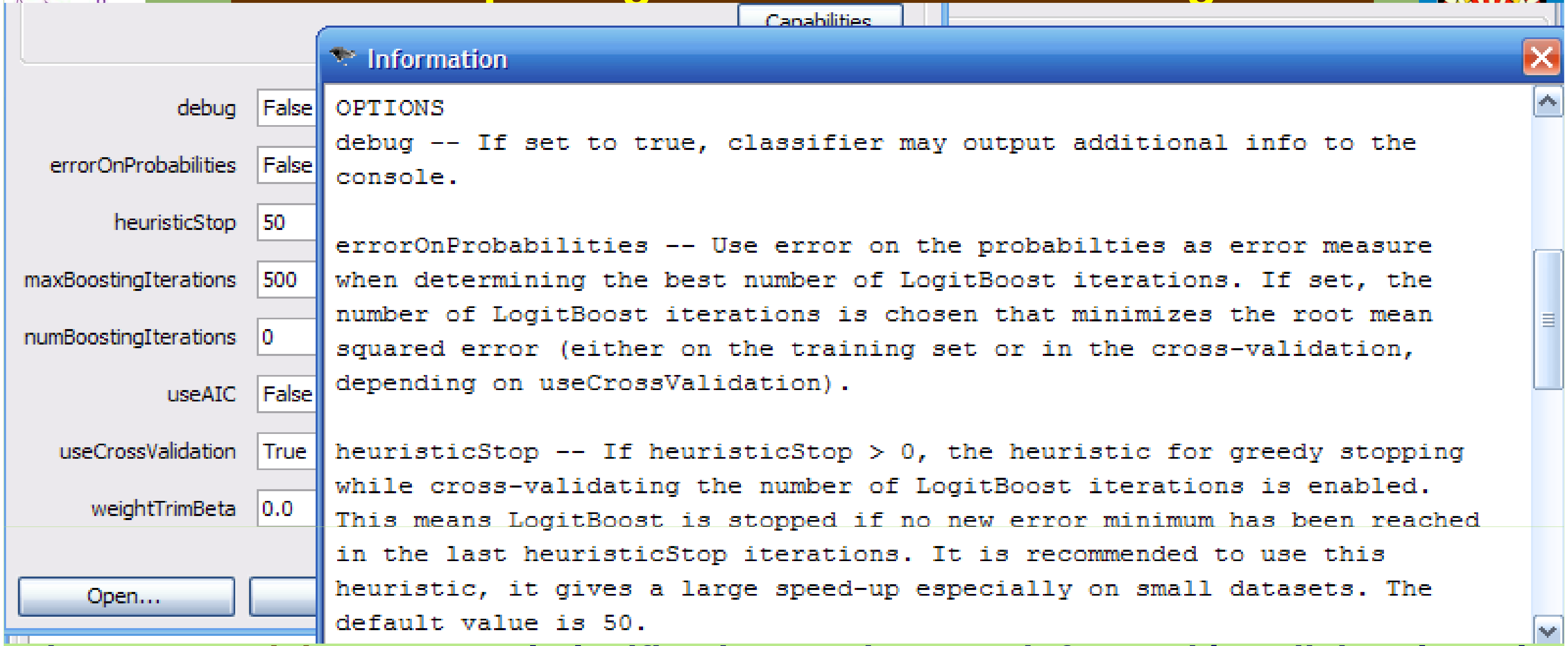
numBoostingIterations -- Set fixed number of iterations for LogitBoost. If >= 0, this sets the number of LogitBoost iterations to perform. If < 0, the number is cross-validated or a stopping criterion on the training set is used (depending on the value of useCrossValidation).

useAIC -- The AIC is used to determine when to stop LogitBoost iterations (instead of cross-validation or training error).

useCrossValidation -- Sets whether the number of LogitBoost iterations is to be cross-validated or the stopping criterion on the training set should be used. If not set (and no fixed number of iterations was given), the number of LogitBoost iterations is used that minimizes the error on the training set (misclassification error or error on probabilities depending on errorOnProbabilities).

weightTrimBeta -- Set the beta value used for weight trimming in LogitBoost. Only instances carrying (1 - beta)% of the weight from previous iteration are used in the next iteration. Set to 0 for no weight trimming. The default value is 0.

SimpleLogistic Weka. Iris training

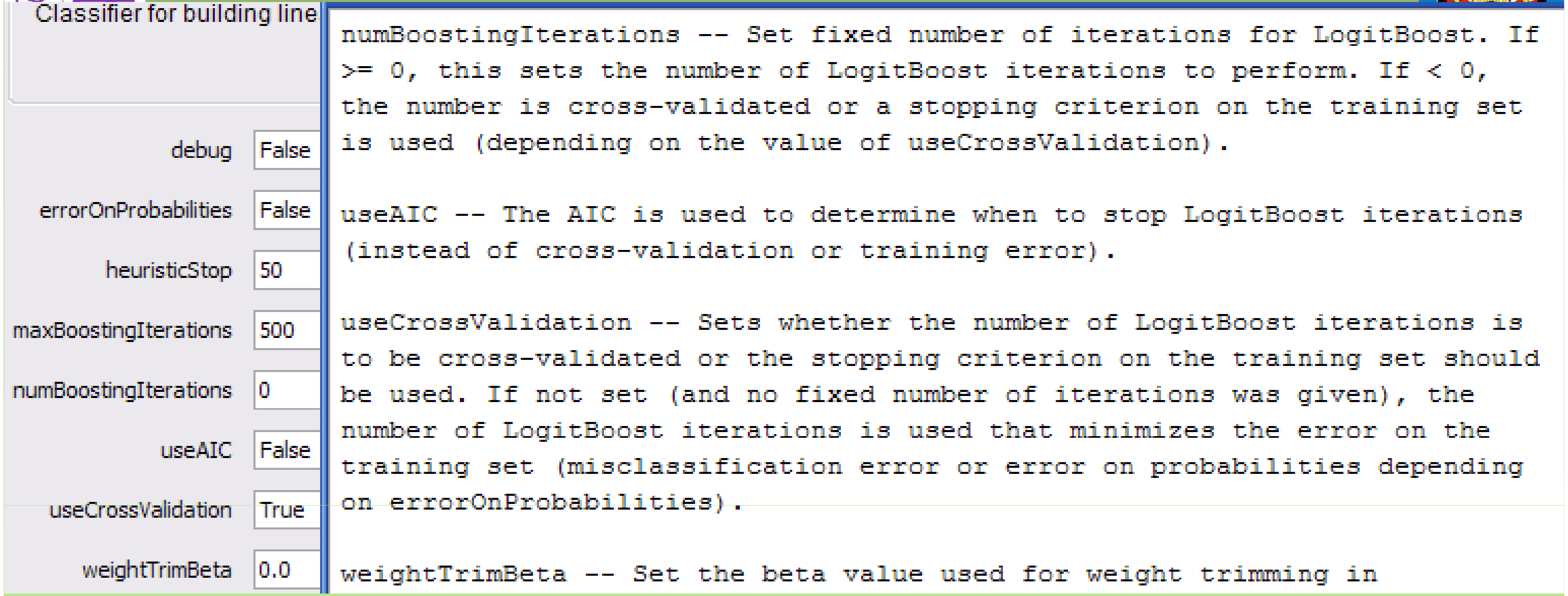


Si ponemos debug a true el clasificador puede sacar información adicional por la consola

Si ponemos errorOnProbabilities a true utiliza el error asociado a las probabilidades como medida de error para determinar el mejor número de iteraciones de LogitBoost, que estará en función de la minimización del RMSE sobre el conjunto de entrenamiento o en el procedimiento de cross-validation

heuristicStop igual a 50 indica que se utiliza un criterio de parada codicioso mientras se crosvalida el número de iteraciones de LogitBoost, esto significa que LogitBoost se para si no se encuentra un error menor en las últimas 50 iteraciones.

SimpleLogistic Weka. Iris training

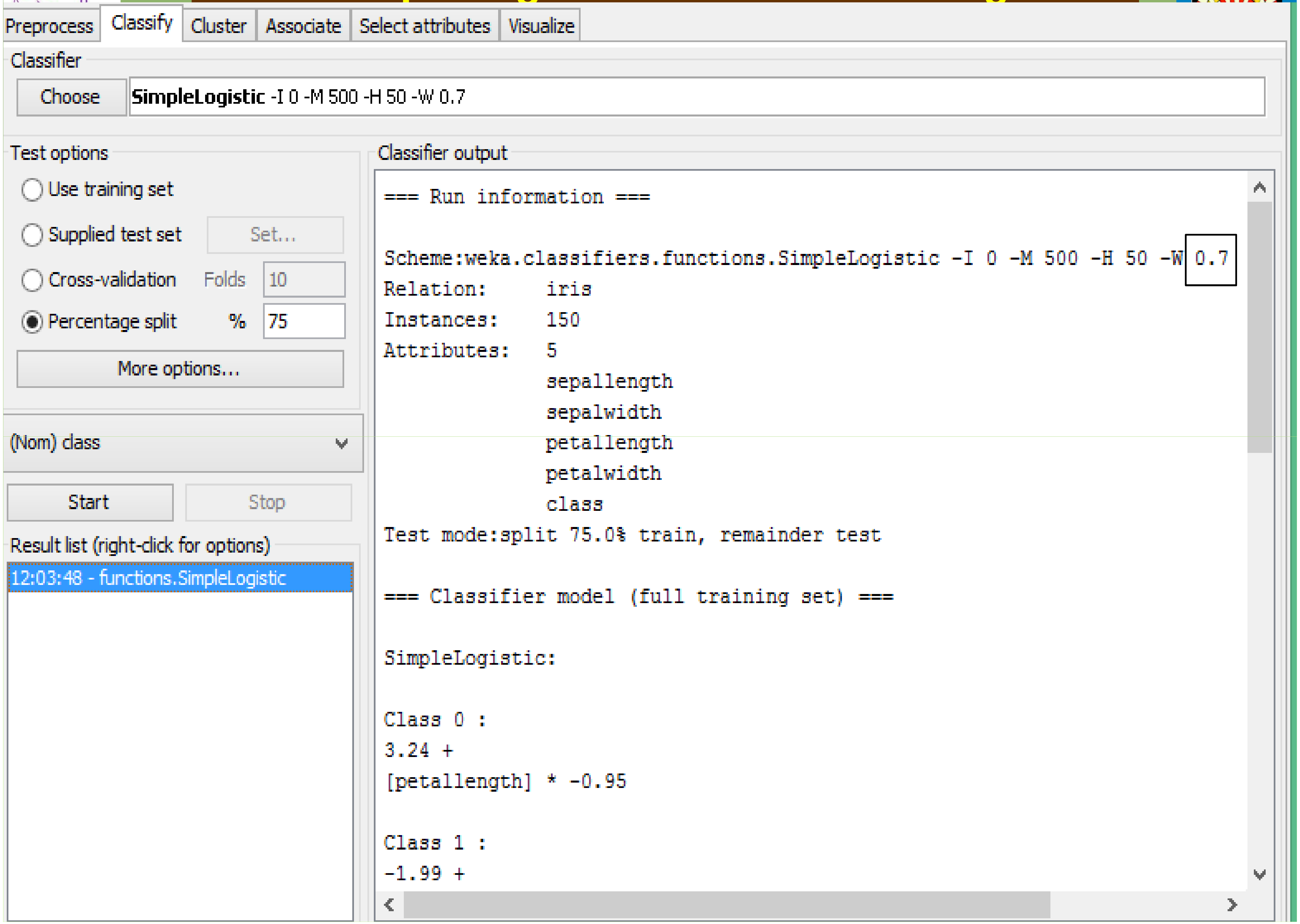


numBoostingIterations define el número de iteraciones para LogitBoost. Si es mayor o igual a 0, define el número de iteraciones a realizar. Si es menor de 0, el número se obtiene por cross-validation o mediante un criterio de parada sobre el conjunto de entrenamiento

UseAIC. Utiliza el Criterio de Información de Akaike para determinar cuando para las iteraciones del LogitBoost en lugar de hacerlo por cross-validation o mediante el error de entrenamiento

UseCroosValidation define si se utiliza este criterio de parada para LogitBoost o si se utiliza el criterio de parada sobre el conjunto de entrenamiento

SimpleLogistic Weka. Iris training



SimpleLogistic Weka. Iris training

```
Class 0 :
3.24 +
[petallength] * -0.95

Class 1 :
-1.99 +
[petallength] * 0.82

Class 2 :
-1.22 +
[petalwidth] * 0.4



Time taken to build model: 0.28 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          31               83.7838 %
Incorrectly Classified Instances         6               16.2162 %
Kappa statistic                          0.756
Mean absolute error                      0.239
Root mean squared error                  0.3143
```

APRENDIZAJE AUTOMÁTICO: TERCER CURSO DEL GRADO DE ING. INFORMÁTICA EN COMPUTACION

Clasificación: Regresión Logística Binaria: Aplicaciones con SPSS y Weka

César Hervás-Martínez
Grupo de Investigación AYRNA

Departamento de Informática y Análisis Numérico
Universidad de Córdoba
Campus de Rabanales. Edificio Einstein.
Email: chervas@uco.es

2019-2020