Diseño y Construcción de Scorecards con Herramientas de Data Mining

Cristián Bravo R.
cbravo@dii.uchile.cl

Banco de Crédito e Inversiones
2 al 5 de Julio, 2011

Agenda del Módulo

- Preparación de datos para generación de scorecards.
  - Selección de Variables.
  - Transformaciones Notables.
  - Segmentación de Universos.
- Modelos avanzados:
  - Análisis de Supervivencia.
  - Redes neuronales en Credit Scoring.
  - Modelos de optimización.
  - SVMs.
  - Consideraciones de Basilea II y uso avanzado de modelos.

Agenda del Módulo

- Inferencia en los Rechazados.
- Construcción de un Scorecard.
  - Transformación a log-odds.
- Stress Testing.
- Puntos de Corte.
- Seguimiento.

Modelos de Credit Scoring: Redes Neuronales en Credit Scoring

Redes Neuronales - Recuerdo

- Método de aprendizaje supervisado.
  - Las redes neuronales pertenecen al conjunto de herramientas de clasificación y regresión no lineal.
  - Se ha demostrado que es un **"aproximador" universal**:
    - Cualquier **función continua acotada** se puede aproximar por una red neuronal con **una** sola capa oculta con cierto nivel de error.
    - Toda **función general** puede ser aproximada por una red neuronal con **dos** capas ocultas.

REDES NEURONALES
Perceptrón Multicapa

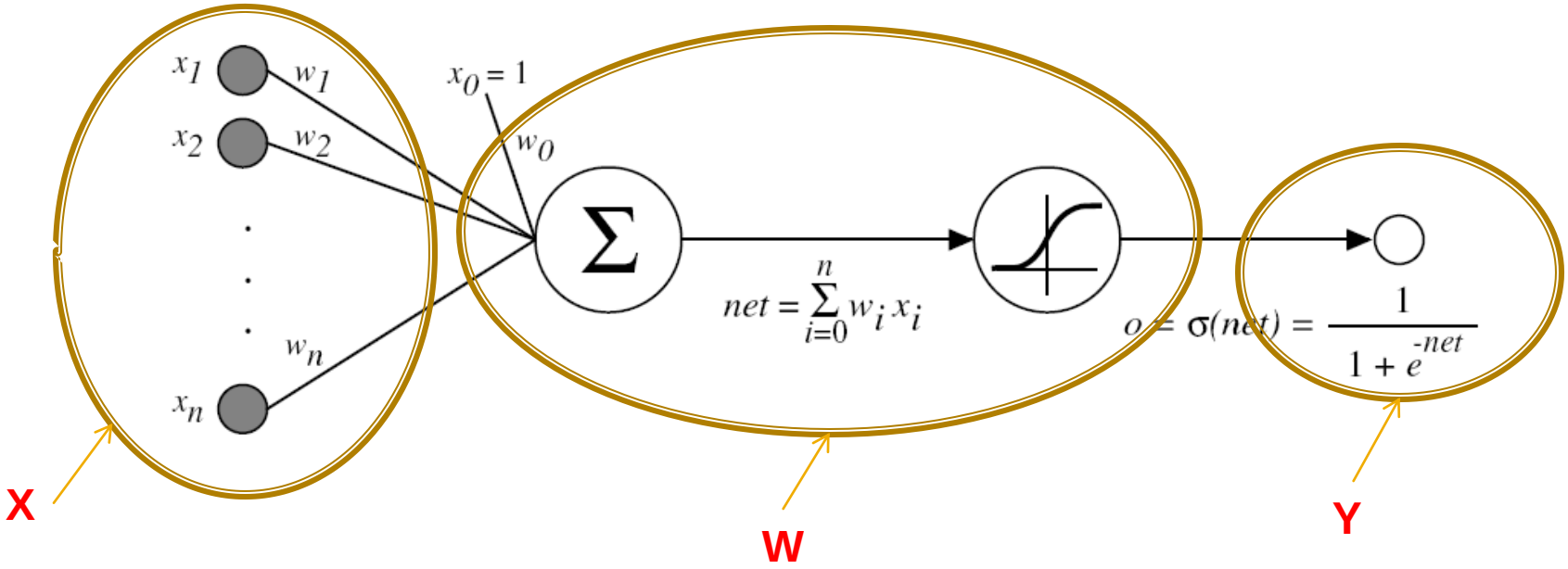
- La arquitectura de la red neuronal se caracteriza porque tiene sus neuronas agrupadas en capas de diferentes niveles.
- Cada una de las capas esta formada por un conjunto de neuronas y se distinguen entre ellas en **3 niveles** de capas distintas:
  - **Capa de entrada:** se encargan de recibir las señales o patrones que proceden del exterior y propagar las señales a todas las otras neuronas de la siguiente capa
  - **Capas ocultas**: son las que tienen la misión de realizar el procesamiento no lineal de los patrones recibidos.
  - **Capa de salida:** actúa como salida de la red, proporcionando al exterior la respuesta de la red, para cada uno de los patrones de entrada.

REDES NEURONALES
Perceptrón Multicapa (II)

- Una **red neuronal** **es una función**

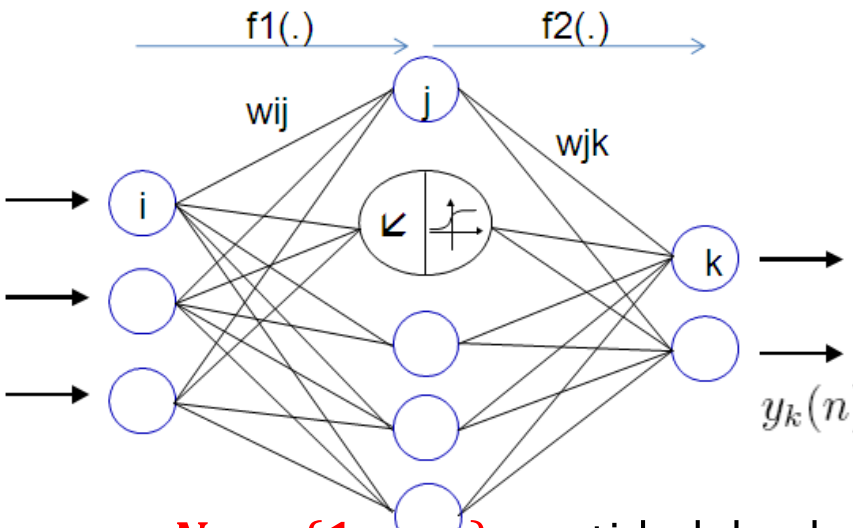
$$Y = F(X, W)$$

donde **Y** es el vector formado por las salidas de la red, **X** es el vector de entrada a la red, y **W** es el conjunto de todos los parámetros de la red (pesos y umbrales), y **F** una función continua no lineal.



Redes Neuronales - Problema

- Se desea resolver el siguiente problema de optimización:



$$\min_{W} E$$

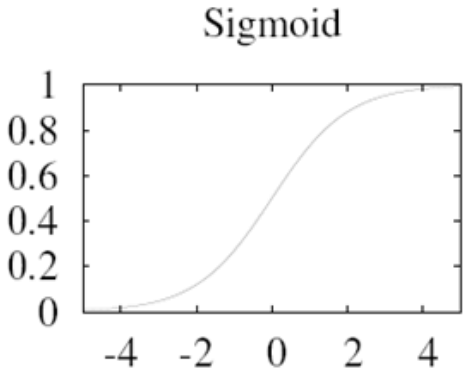
$$E = \frac{1}{N} \cdot \sum_{n=1}^{N} \left(\frac{1}{2} \sum_{k=1}^{nC} \left(s_k(n) - y_k(n) \right)^2 \right)$$

$$y_k(n) = f_2 \left(\sum_{j=1}^{nc} w_{j,k} \cdot f_1 \left(\sum_{j=1}^{nc} w_{i,j} \cdot x_n^i + u_j^c \right) + u_k^c \right)$$

- $N = \{1, .., n\}$ cantidad de observaciones
- n_C = cantidad de neuronas de salida (variable objetivo)
- x_n^i = valor del atributo **i** para la observación **n**
- $S_k(n)$ = valor real de la clase para la observación **n**
- $Y_k(n)$ = valor calculado para la observación **n.**
- $E =$ Error de la red.

Funciones de Salida

- Sigmoidal: Para salidas estrictamente positivas.



$$y(x) = \frac{\alpha}{1 + e^{-\beta \cdot x}}$$

  - α: Factor de escala. Función entrega valores entre 0 y 1, por lo que se debe escalar.
  - Solvers usualmente traen implementada esta función sólo entre [0,1]. Solución: normalizar objetivo.
  - Es la más razonable para probabilidades.
  - En caso de probabilidades multiclase, usar **softmax**.

El Problema de la Caja Negra

- Las redes neuronales son **cajas negras**
  - La salida de los parámetros no tienen una interpretación sencilla, ni directa.
  - Al existir complejas interrelaciones no lineales, obtener el efecto de una variable de forma lineal simplemente no es factible.
- Esto trae una serie de problemas regulatorios.
  - No existe forma directa de determinar en qué factores el cliente posee riesgo alto.
  - Existen algunas maneras experimentales de aplicar el modelo.

Modelo Bravo *et al.* 2010

- La primera forma, sencilla, de adaptar el proceso es a través de una red neuronal que posea una configuración lineal en la salida.
- Se diseña de la siguiente manera:
  - Función de transferencia: lineal.
  - Función de salida: sigmoidal.
  - Función de error: Entropía cruzada.

$$E = \sum_{n} S_n \ln(y_n(x)) + (1 - S_n) \ln\big(1 - y_n(x)\big)$$

  - Con esta configuración se entrena un paralelo de regresión logística y máxima verosimilitud, pero manteniendo las propiedades de aproximación de las redes neuronales.

Modelo Bravo *et al.* 2010 (II)

- Bajo esta configuración se obtienen los siguientes parámetros:
  - Matriz de entrada U de tamaño $(V+1) \times (H+1)$.
    - H: Cantidad de neuronas en capa oculta.
  - Matriz de transferencia T, de tamaño $(H+1) \times 1$.
- Es posible entonces calcular una probabilidad sigmoidal que posee los siguientes valores:

$$p(x) = \frac{1}{1 + \exp(-U \times T \times x)}$$

  - Esta es una **probabilidad lineal**, sobre la cual se puede estimar un score y analizar de forma clásica.

Modelo de Waal *et al.* 2008

- Otra forma es plantear un **modelo lineal generalizado**, y utilizar este para obtener una respuesta de cada variable de forma no lineal.
- Planteamos ahora la función de regresión como:

$$y(x_i) = \beta_0 + \sum_k f_k(x_{ik})$$

  - Con β_0 un sesgo y f_k una función general (no lineal).
  - Esta función, al ser aproximada por una red neuronal, se le llama "Red Neuronal Aditiva General" (GANN).
  - Permite la generación de scores no lineales.

Arquitectura GANN

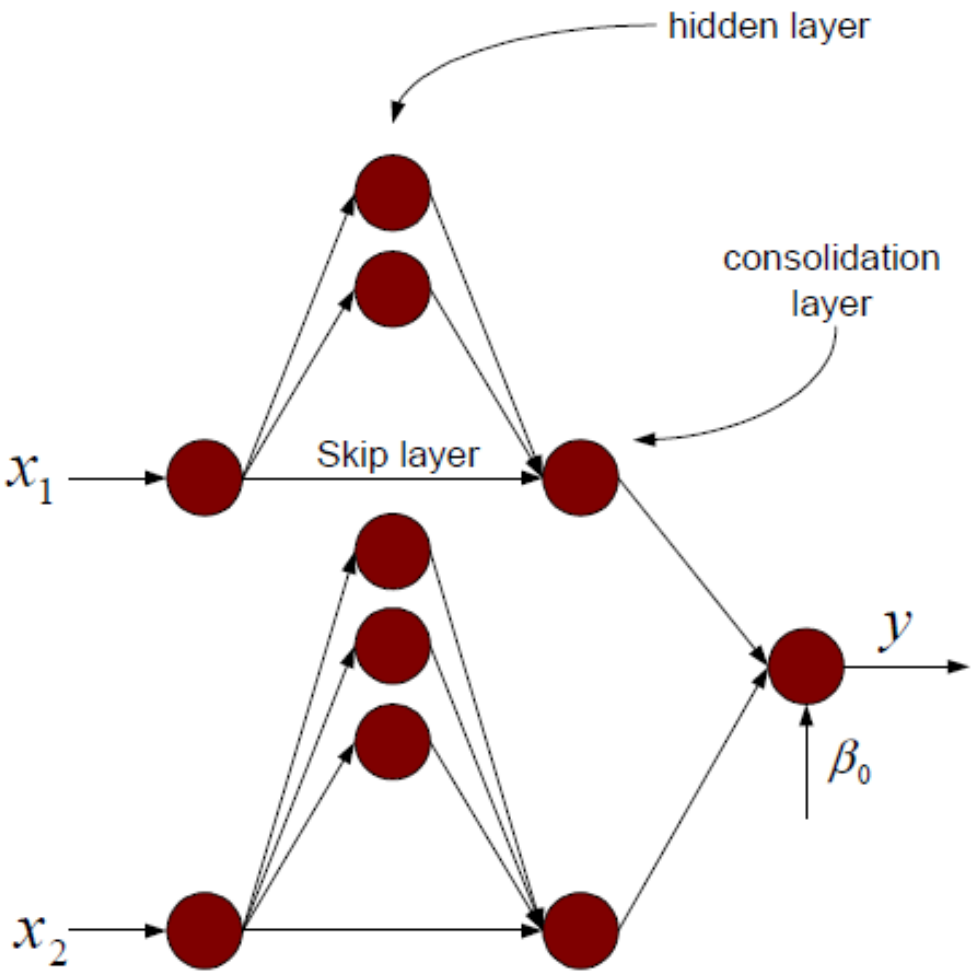


Figure 1: Example GANN

Función de Transferencia GANN

- En principio cualquier función de transferencia regular puede ser utilizada, en particular los autores proponen:

$$f_j(x_{ji}) = w_{1j}\tanh(w_{01j} + w_{11}x_{ji}) + \cdots$$
$$+ w_{hj}\tanh(w_{0hj} + w_{1hj}x_{ji})$$

- Esta función entrega respuestas entre -1 y 1.
  - Otra opción es usar sigmoidal en capa oculta. Para datos en [0,1] esto mejora resultados.

Efecto de cada Variable en GANN: Gráfico de Residuos Parciales

- Ahora es necesario generar una medida que indique qué factores son los más importantes.
- El **gráfico de residuos parciales** permite identificar relaciones no lineales entre distintas variables asociadas a un modelo.
- El procedimiento requiere haber ajustado el modelo general y encontrado el estimador para cada variable.

Efecto de cada Variable en GANN: Gráfico de Residuos Parciales (II)

- El residuo parcial de una variable j y un caso i será:

$$pr_{ij} = g_0^{-1}(y_i) - \beta_0 - \sum_{l \neq j} f_l(x_{li})$$

  - g_0^{-1}: Inversa de la función de activación en la capa de salida de y (inversa de sigmoidal).
- El gráfico de estos residuos contra el valor de la variable indica cuán relevante es la variable en el modelo general.

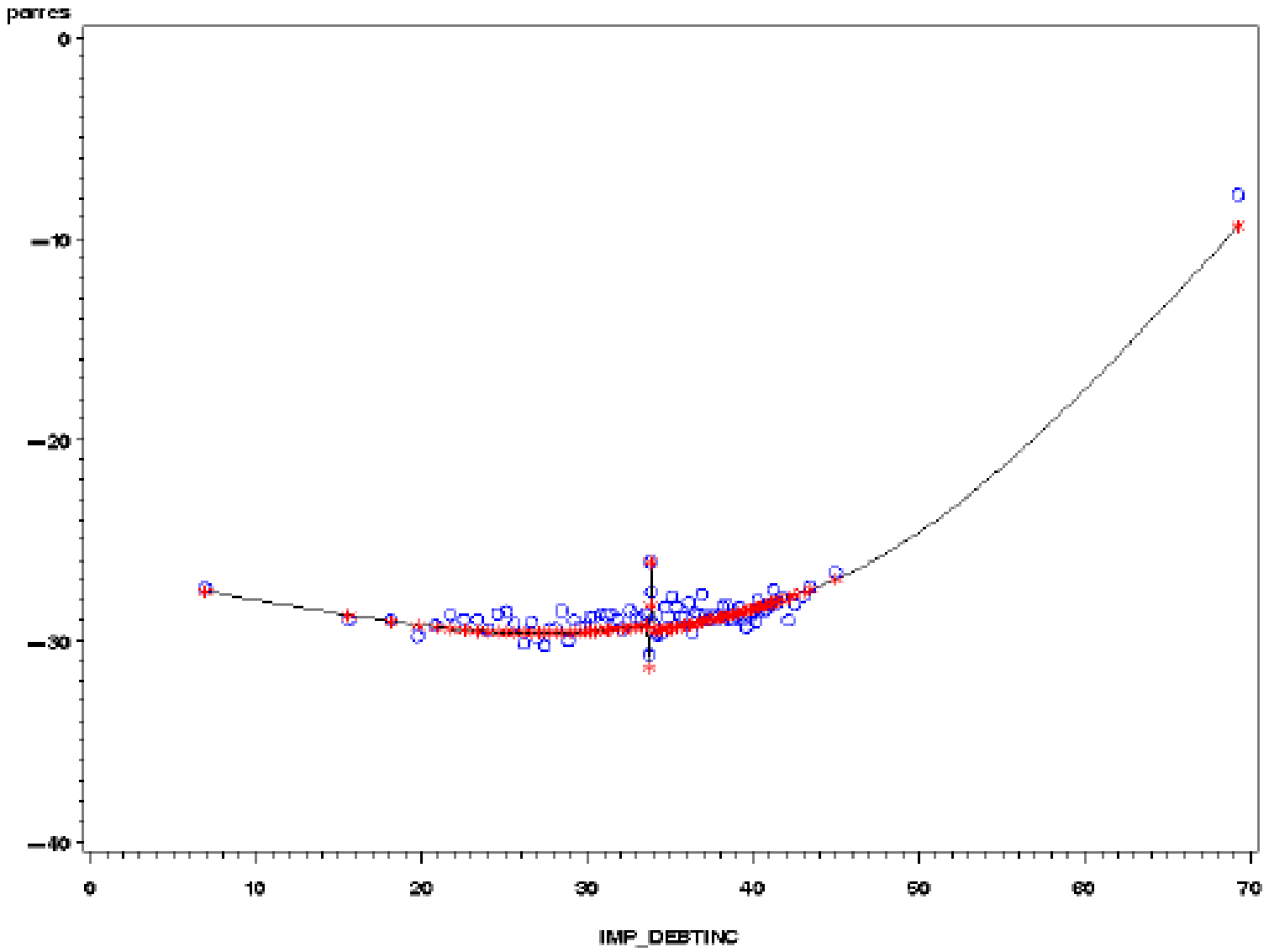
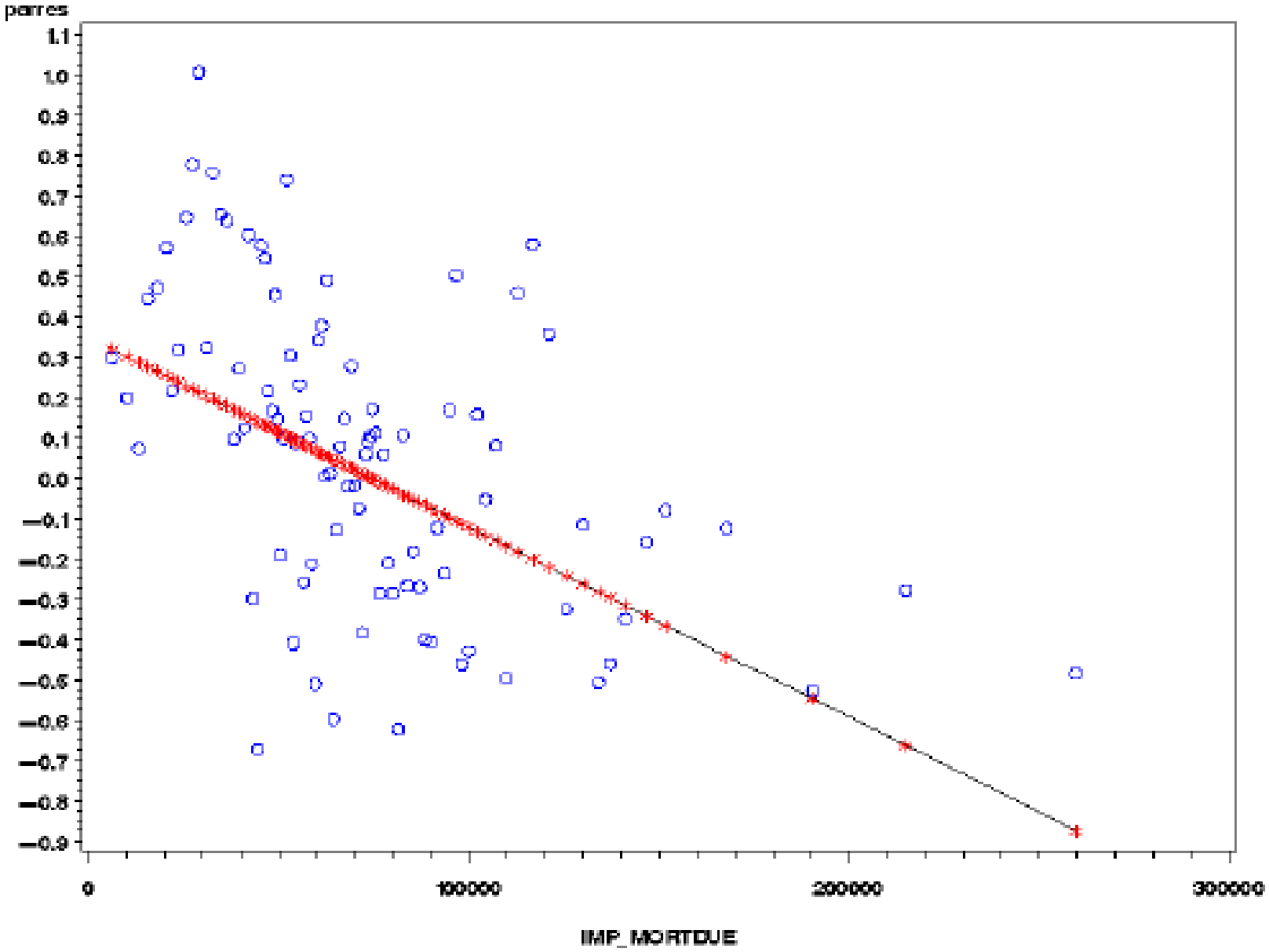


Figure 5: Partial Residual Plot for IMP_DEBTINC


parres
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
-0.8
-0.9
0
100000
200000
300000
IMP_MORTDUE

Seleccionando los Parámetros de las Redes Neuronales

- En general, el procedimiento es siempre dividir un porcentaje de la muestra independiente para obtener parámetros.
  - Ej: 20% para parámetros, 60% para entrenamiento, 20% para test independiente.
- Es importante que la muestra sea independiente, pues es **parte activa del entrenamiento**.
- En la muestra para obtener parámetros se utiliza ahora una **búsqueda en grilla**.

Seleccionando los Parámetros de las Redes Neuronales.

- El proceso iterativamente determina los mejores parámetros en una **muestra independiente**.
  1. Determinar rangos posibles de parámetros en potencias de 10 (ej: $[10^{-4}, 10^{4}]$).
  2. Probar los parámetros usando pasos que cubran el espectro con potencias (avanzar de a uno en exponente).
  3. Una vez determinado el mejor, probar en ese intervalo a intervalos de un factor menor.
     - Ej: Si mejor valor es 10, probar ahora entre 1 y 100, de a pasos de a 1.
  4. Elegir mejor valor.

Modelos de Credit Scoring: Optimización Lineal y SVMs

Support Vector Machines Recordatorio

- Potente método estadístico basado en hiperplanos.
- Propiedad:
  - Minimiza "Riesgo Estructural", es decir, el riesgo de aproximar tanto los casos que observas en la muestra como los que no.
- ¡Hiperplanos son lineales!
  - Idea: Utilizar Kernels para poder modelar fenómenos no lineales.

Support Vector Machines
Formulación del problema (1)

- Formulación **Primal** del Problema:

$$\begin{aligned} \text{mín} \quad & \frac{1}{2}\sum_{i=1}^{n} w_i^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i\left(\langle w^T, x_i\rangle + b\right) \geq 1 - \xi_i \\ & \xi_i \geq 0, \forall i = 1, ..., n \end{aligned}$$

- Formulación **Dual** del Problema:

$$\begin{aligned} \text{máx} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle \\ \text{subject to} \quad & C \geq \alpha_i \geq 0, \forall i = 1, ..., n \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned}$$

Support Vector Machines
Propiedades del parámetro C

- El parámetro C es quien acompaña al objetivo de minimizar el error ocasionado por la maximización del margen.
- Representa el **trade-off** entre el error de clasificación y la complejidad del modelo.
  - Valores grandes de C favorecen soluciones con pocos errores de clasificación.
  - Valores pequeños de C expresan preferencias a modelos con menor complejidad.
- El valor de C se considera como el parámetro de regularización en las SVMs.

Función Kernel
Definiciones

- Definimos la función de "mapeo":

$$\begin{aligned} \phi: \quad & \mathbb{R}^n \rightarrow F \\ & x \rightarrow \phi(x) \end{aligned}$$

- Definimos la función Kernel:

$$k(x, x') := (\phi(x) \cdot \phi(x'))$$

$$\begin{aligned} k: \quad & \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} \\ & (x, x') \rightarrow k(x, x') \end{aligned}$$

Función Kernel
Definiciones (2)

- Actualizamos la formulación en el producto punto del problema dual por la función Kernel:

$$\begin{aligned} \text{máx} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ \text{subject to} \quad & C \geq \alpha_i \geq 0, \forall i = 1, ..., n \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

Función de Decisión

- Para utilizar la función de decisión, en el caso lineal se calcula:

$$f(x_j) = sgn\left[\sum_{i \in SV} \alpha_i y_i < x_i, x_j > + b\right]$$

- En el caso no lineal:

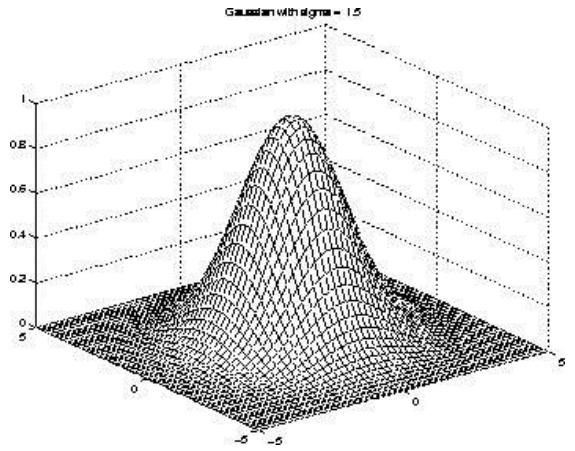
$$f(x_j) = sgn\left[\sum_{i \in SV} \alpha_i y_i k(x_i, x_j) + b\right]$$

  - SV: Vectores soportantes de la SVM (del entrenamiento).
  - α_i: Variable dual vector soportante.

Función de Kernel
Ejemplos

- "**Radial Basis Function**" o Kernel Gaussiano.

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right)$$



- Propiedades
  - **Dimensión infinita**: Todo es separable por un hiperplano.
  - Parámetro permite ajustar curvatura de los datos.

Usando SVMs en Credit Scoring: Preparando la Muestra

- SVM maneja bien todo tipo de datos, tanto continuos como discretos.
  - Es necesario que todos los datos estén en la misma escala ($[0,1]$ por ejemplo).
- La base de datos no tiene que estar balanceada necesariamente.
  - SVM posee el concepto de "balanced ridge" que multiplica a los casos de la clase con menos casos.
    - Este valor debe ser de $\frac{\#Malos}{\#Buenos}$.
- Ahora se debe realizar una **búsqueda en grilla** para obtener los parámetros.

Usando SVM en Credit Scoring: Interpretabilidad

- En el caso de kernel lineal, el vector w si entrega un clasificador con pesos por cada variable, que se puede usar para analizar efecto **por cliente**:

$$w = \sum_{j \in SV} \alpha_j y_j x_j$$

- En el caso no lineal, al igual que en redes neuronales, el problema en SVM es la **interpretabilidad** de los parámetros.
  - No es posible saber a ciencia cierta qué parámetros afectan más a cada cliente en particular.
  - Si es posible, sin embargo, detectar aquellas variables que tienen mayor efecto en el modelo.

Pseudo-W de Guyon (2002)

- La metodología es obtener el vector w asociado a la SVM no lineal.
  - Como el kernel es no lineal, el vector w corresponde a:

$$w = \sum_{j \in SV} \alpha_j y_j \phi(x_j)$$

  - Si $\phi(x_j)$, la transformación al espacio nuevo, no es obtenible de forma directa, se aproxima de forma numérica.
  - Variables con un mayor w tendrán un mayor peso en el modelo.

Comparación de Eficiencia

- La eficiencia del modelo fue comparada por Belloti & Cook (2007) con una serie de otros modelos en un gran dataset (25.000 créditos).

Classification algorithm	*Test AUC (mean)*	*Standard deviation*
LR	0.779	0.0063
LR with interaction variables	0.777	0.0076
SVM: linear	0.783	0.0055
SVM: polynomial	0.755	0.0068
SVM: Gaussian RBF	0.783	0.0053
LDA	0.781	0.0058
*k*NN	0.756	0.0063

Probabilidades en SVM - Método de Platt

- SVM entrega salida binaria.
- ¿Cómo calcular probabilidades continuas?
- Método de Platt.
  - Metodología para calcular probabilidades a través de salida continua de SVM.
  - Input: función de clasificación **sin función signo (continua)**.

$$f(x) = \sum_{j \text{ es } SV} \alpha_j y_j k(x, x_j) + b_j$$

Probabilidades en SVM - Método de Platt (II)

- Estima función de probabilidad dada por:

$$p(x) = \frac{1}{1 + \exp\left(A \cdot f(x) + B\right)}$$

- Parámetros A y B se estiman utilizando máxima verosimilitud.
- Método supone una **distribución sigmoidal** para la función p.
- Si el kernel es lineal, tenemos una función sobre la que se puede armar un score.

Inferencia Sobre los Rechazados

Inferencia sobre los Rechazados

- Una vez que se tiene el scoring preliminar, queda la opción de **incorporar las solicitudes que fueron rechazadas**.
- Problema: Estas solicitudes NO poseen variable objetivo.
- Solución: Existen varias posibles.
  - Ignorarlas: Se tiene población mal representada.
  - Considerarlas a todas como buenas o malas: Patrón cambia considerablemente. Supuesto no sustentable.

Inferencia sobre los Rechazados (II)

- ¿Cuándo hacer inferencia?
  - NO hacer inferencia si:
    - Se tiene confianza en el proceso de aceptación/rechazo inicial.
    - Si se aceptan una alta cantidad de solicitudes contra las que se rechazan.
  - Hacer inferencia si:
    - Se tienen altas tasas de rechazo.
    - Se tienen pocos casos malos en la base de datos original.

Técnicas de Inferencia: Parcelamiento

- Técnica sencilla, basada en el scoring inicial.
  - Ej: Score entre 0 y 300.

Score	# Bad	# Good	% Bad	% Good	Reject	Rej - Bad	Rej - Good
0–169	290	971	23.0%	77.0%	1,646	379	1,267
170–179	530	2,414	18.0%	82.0%	1,732	312	1,420
180–189	365	2,242	14.0%	86.0%	3,719	521	3,198
190–199	131	1,179	10.0%	90.0%	7,334	733	6,601
200–209	211	2,427	8.0%	92.0%	1,176	94	1,082
210–219	213	4,047	5.0%	95.0%	3,518	176	3,342
220–229	122	2,928	4.0%	96.0%	7,211	288	6,923
230–239	139	6,811	2.0%	98.0%	3,871	77	3,794
240–249	88	10,912	0.8%	99.2%	4,773	38	4,735
250+	94	18,706	0.5%	99.5%	8,982	45	8,937

  - Inferencia se hace a través de asignar la proporción observada al caso.
    - Puede modificarse según sea necesario.

Técnicas de Inferencia: Reclasificación Iterativa

- Método basado en **búsqueda greedy**.
- Pasos:
  1. Construir scorecard con datos conocidos.
  2. Asignar solicitudes rechazadas a casos buenos o malos acorde a algún puntaje base (criterio).
  3. Combinar los casos inferidos con los asignados y construir nueva base de datos con todos los casos.
  4. Volver a 1.
     - **Criterio de parada**: Cuando no existe cambio en las solicitudes rechazadas entre una iteración y otra.

Re-Modelamiento y Verificación de la Inferencia

- Una vez lista la inferencia, se construye el modelo de scoring nuevamente.
- Concluido éste, es necesario verificar que los **casos están bien clasificados**.
- Se debe verificar:
  - Que los casos clasificados nuevos sean razonables.
  - Que los WOE de las variables en el modelo nuevo **tengan sentido**.

Escalamiento a Score

Creación de Score Final

- Ahora es necesario **reconstruir el modelo**.
  - Dependiendo de la técnica de inferencia, éste podría estar terminado ya.
  - Nota: NO es necesario mantener las variables del modelo inicial.
- Hasta este momento tenemos una salida continua, en el intervalo [0,1].
- Muchas veces, tener este score no es deseable, es preferible tener un puntaje más homogéneo.
  - La configuración real del modelo de scoring es secreto comercial.
  - Es deseable tener una **transformación no invertible** del score.

Escalamiento a Score

- Un score debe cumplir qué:
  - El puntaje se encuentre en una escala más comprensible. (Ej: 1-1000)
  - Cada rango de atributo tendrá asociado un score positivo o negativo. (Ej: Si edad entre 20 y 30, sumar 10)
  - El score final debe ser la suma de los scores particulares.

Escalamiento a Score (II)

- El score más utilizado es el score logarítmico.

$$Score = Escala + \ Factor * \ln(odds)$$

- La definición requiere recordar los siguientes conceptos:
  - Odds: Chance de ocurrencia de un evento. Ej: 50:1 implica que uno de cada 50 usuarios será mal pagador.
  - En la regresión logística, los odds corresponden a la suma de las variables por su coeficiente beta, multiplicado por -1.

Escalamiento a Score (III)

- Definimos:
  - Puntos para Duplicar las Odds (PDO): Cantidad de puntos necesaria para duplicar las chances de ocurrencia del evento.
- Entonces, el score debe cumplir que:

$$Score = Escala + \ Factor * \ln(odds)$$

$$score + pdo = Escala + factor * \ln(2 * odds)$$

$$\ln(odds) = -(\sum \beta_i * x_i + \beta_0)$$

Escalamiento a Score (IV)

- Resolviendo el sistema anterior se tiene que:

$$Factor = \frac{pdo}{\ln(2)}$$

$$Escala = Score - Factor * \ln(Odds)$$

$$\ln(odds) = -(\sum \beta_i * x_i + \beta_0)$$

- Ejemplo:
  - Pdo = 20 puntos.
  - En puntaje 600 quiero que se tengan chances de 50:1

Escalamiento a Score (V)

- Resolviendo lo anterior:

$$Factor = \frac{20}{\ln(2)} = 28.85$$

$$Escala = 600 - 28.85 * \ln\left(\frac{50}{1}\right) = 487.12$$

- Así el score final resulta:

$$Score = 487.12 - 48.85 * (\sum \beta_i * x_i + \beta_0)$$

Uso del Score

- Una de las ventajas del score es que permite dar una respuesta al cliente acerca de las razones para el rechazo.
- Necesitamos:
  - Score por variable.
  - Score base por variable.
- Supongamos que existen V variables, con k_v categorías cada una.

Uso del Score (II)

- El score corresponde a:

$$score = escala - factor * \left(\sum_{V,k_v} \beta_{k_V} x_{k_v} + \beta_0 \right)$$

$$score = escala - \sum_{V} \left(factor * \left(\sum_{k_v} \beta_{k_V} x_{k_v} + \frac{\beta_0}{V} \right) \right)$$

$$Score = \sum_{V} \left(\frac{Escala}{V} - factor * \left(\sum_{k_v} \beta_{k_v} * x_{k_v} + \frac{\beta_0}{V} \right) \right)$$

- El score neutro corresponde a eliminar todas las variables.

Uso del score (III)

- Score Neutro:

$$Score\ Neutro = \frac{Escala}{V} - factor * \frac{\beta_0}{V}$$

- La idea es entonces calcular el valor del puntaje, por variable, para los demás casos.
- Toda variable que tenga un valor de score **por debajo del score neutro**, tiene asociado un **riesgo alto**.