El modelo presentado en este trabajo, teóricamente pesado y con una implementación tediosa que recae en técnicas de simulación, resultó ser realmente efectivo en la práctica. A lo largo de este capítulo, se hará una exploración intuitiva y muy visual de sus capacidades. Todas las gráficas presentadas, se generaron con el mismo paquete que realiza la estimación, pues los mismos objetos que las funciones de `R` arrojan, pueden ser utilizadas para hacer gráficas que reflejan la intuición subyacente del modelo.

En particular, se simularon cinco bases de datos en dos dimensiones, es decir, se tienen dos covariables $\mathbf{X} \in \mathbb{R}^2$, con diferentes patrones para la respuesta y tanto lineales como no lineales. Esto, con el objetivo de poder *visualizar* los grupos, como se separan por medio de fronteras no lineales y para poder visualizar la función $f(\mathbf{x})$ en tres dimensiones. Posteriormente, se aplica a bases de datos reales, particularmente una de cáncer y otra financiera, donde, al aumentar la dimensionalidad no se pueden visualizar. Sin embargo, se dan una serie de resúmenes numéricos y medidas que evalúan la precisión del modelo, abriendo la discusión a limitaciones de este.

0.1. Evalución del modelo - función log-loss y matrices de confusión

Dos buenas medidas de evaluar la efectividad (y precisión) de un modelo de clasificación binaria, son las *matrices de confusión* y la función *log-loss* (LL).

Sea $\mathbf{y} = (y_1, \ldots, y_n)^t$ el vector de respuestas verdaderas; $\hat{\mathbf{p}} = (\hat{p_1}, \ldots, \hat{p_n})^t$ el vector de probabilidades ajustadas, donde $\hat{p}_i = \hat{P}_{\text{modelo}}(y_1 = 1|\mathbf{x}_i)$ es la probabilidad estimada por el modelo de que la observación y_i sea igual a 1, definiendo el vector de respuestas ajustadas $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n)^t$, donde, $\hat{y}_i = 1 \iff \hat{p}_i > .5$. La función log-loss $ll : \{0, 1\}^n \times [0, 1]^n \to \mathbb{R}^+$ es:

$$ll(\mathbf{y}, \hat{\mathbf{p}}) = -\sum_{i=1}^{n} [y_i \ln(\hat{p_i}) + (1 - y_i) \ln(1 - \hat{p_i})] \tag{1}$$

La ventaja de usar la función ll, es que da una métrica que, no solo para mide que tan buena es la clasificación binaria, sino, que toma en cuenta la precisión de la predicción. Esto se debe a la función es convexa y se penaliza cuando las probabilidades ajustadas están muy lejos de la real. Asimismo, si la predicción fue incorrecta pero la probabilidad fue cercana a 0.5 no se penaliza tanto. Idealmente $ll = 0$ si se da una clasificación perfecta y conforme crezca, el modelo es peor. En la práctica y bajo un enfoque frequentista, la función LL es la que usualmente se utiliza para entrenar y comparar modelos de clasificación como redes neuronales.

El segundo método, la matriz de confusión, no es más que un método descriptivo, con base en *tablas de*

contingencia que calcula las frecuencias para aciertos y errores, separando en grupos. Esto es:

	$\hat{y}=0$	$\hat{y}=1$	
$y=0$	#0's ✓	#0's clasificados como 1	# de observaciones cero
$y=1$	#1's clasificados como 0	#1's ✓	# de observaciones uno
	# de estimados cero	# de estimados uno	Total de obs. $= n$

Tabla 1: Matriz de confusión

De donde se puede ver la exactitud en las predicciones del modelo. Estos dos métodos, serán los usados para evaluar cada ejemplo.

0.2. Análisis a fondo de una modelo sencillo

El primer ejemplo que se analizará, es un ejemplo muy sencillo, que busca ejemplificar cada componente del modelo. Se simularon un total de $n = 350$ observaciones separadas en dos grupos, cada uno con tamaños $n_0 = 200$ y $n_1 = 150$ respectivamente ($n = n_0 + n_1$). Los puntos se muestraron de distribuciones normales bivariadas, esto es:

$$\text{Grupo 0: } \mathbf{x}_i \sim N_2\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| \mu_0 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \Sigma_0 = \begin{pmatrix} 0.25 & 0.35 \\ 0.35 & 1 \end{pmatrix}\right) \qquad i = 1, \ldots, 200$$

$$\text{Grupo 1: } \mathbf{x}_i \sim N_2\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| \mu_1 = \begin{pmatrix} 4 \\ 4 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 1 & .24 \\ .24 & .64 \end{pmatrix}\right) \qquad i = 201, \ldots, 350$$

Las medias μ se toman diferentes para dar una clara separación y las covarianzas corresponden a las correlaciones $\rho_0 = 0.7$ y $\rho_1 = .3$ respectivamente. Codificando el grupo 0 con $y = 0$ de color rojo y el grupo 1 con $y = 1$ de color azul. Se tienen los datos presentados en la Figura 1. Los parámetros se escogieron con un proceso de *prueba y error* para dar estructura pero a la vez separación en el espacio de covariables $\mathcal{X}^2 \approx [0.3, 7.5] \times [-0.5, 5.9]$. Esto, para que se tuviera una pequeña región donde las distribuciones se traslaparan y exista cierto grado de confusión. El objetivo del modelo, es poder hacer una separación de estas dos regiones sin sobreajustar; identificando, a grandes rasgos, dónde se encuentran los puntos rojos y dónde se encuentran los puntos azules.

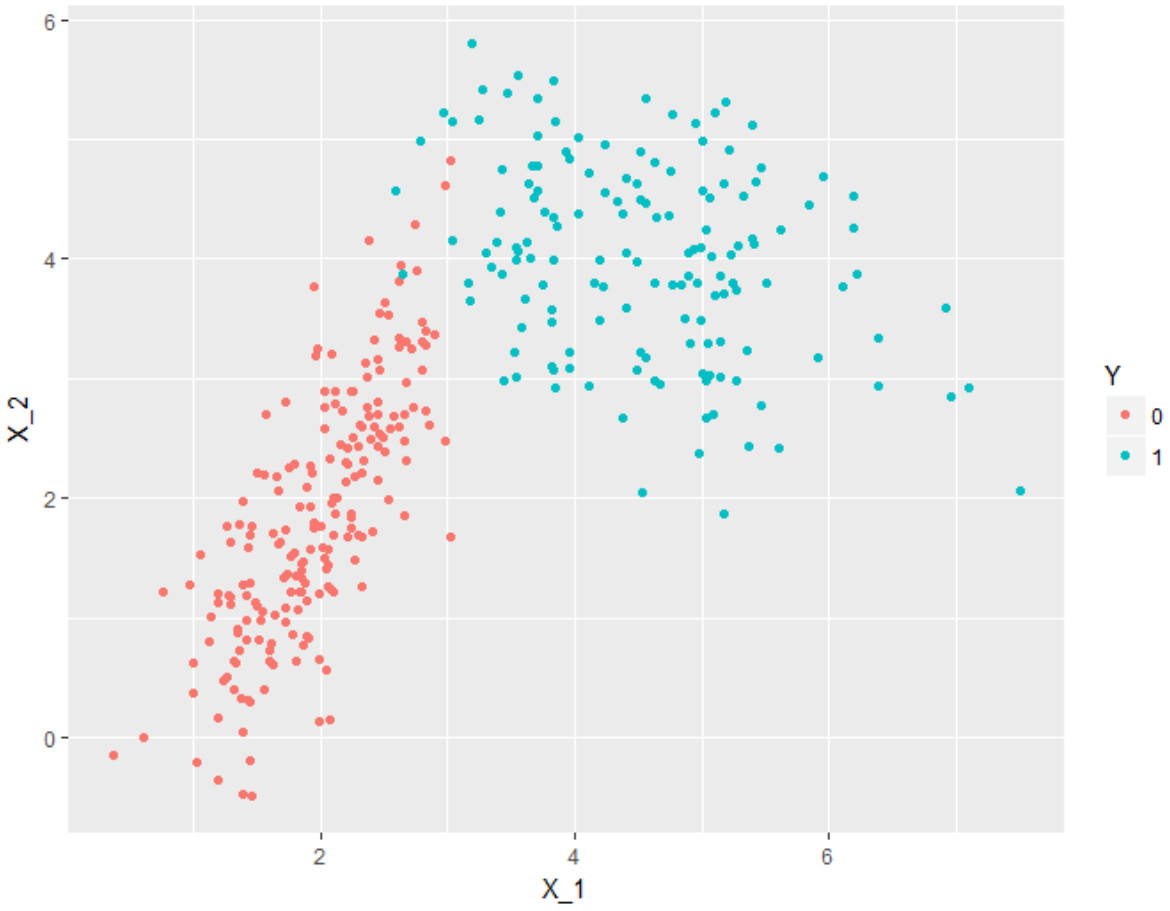


Figura 1: **Primer ejemplo**. Poco traslape entre puntos

Modelo probit frecuentista para comparar

En un modelo tradicional, solo se puede hacer una separación lineal. Para comparar, se corrió el siguiente modelo probit frecuentista en `R`, usando la función `glm(..., family = binomial(link = 'probit'))`:

$$\begin{aligned} p_i = P(y_i = 1) = \mathbb{E}[y|\mathbf{x}_i] = \Phi(f(\mathbf{x}_i)) \quad &\Rightarrow \\ \Phi^{-1}(p_i) = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} \qquad \forall i = 1, \ldots, n \end{aligned} \tag{2}$$

De donde se obtuvieron los siguientes resultados:

Parámetro	Valor Estimado
$\hat{\beta}_0$	-17.29
$\hat{\beta}_1$	4.43
$\hat{\beta}_2$	1.08
Métricas	**Valor**
ll	0.0399

	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	198	2	200
$y = 1$	2	148	150
	200	150	350

Dada la simplicidad de los datos, el modelo lineal probit, presentado en la ecuación (2) resulta ser una excelente forma de hacer la clasificación. Como se ve en la Figura: 2, los datos son fácilmente separables por una linea recta que cruza exactamente donde se empiezan a traslapar. Únicamente, existen 4 datos que quedan mal clasificados, pero que, dadas sus coordenadas, parecerían pertenecer a los grupos opuestos y se consideran como datos atípicos. El modelo presenta una precisión de 98.85 %, todos los parámetros fueron significativos[1] y se tiene un valor de la función log-loss muy bajo. Por lo tanto, se puede concluir que el modelo probit tradicional, es un muy buen modelo para este conjunto de datos.

Figura 2: Separación de los grupos por medio de un modelo probit lineal frecuentista

Primer ejemplo del modelo bpwpm

Ahora, se prueba contra el modelo *bpwpm* de este trabajo. Se esperaría que, al menos, se comportara como el probit anterior, pues, dada la estructura tan robusta que se tiene, este podría colapsar en uno lineal.

Como un primer ejemplo sencillo de comprender, se corren un modelo donde los polinomios por partes son rectas disjuntas en cada segmento con un solo nodo. Se realizan mil simulaciones del muestreo Gibbs, de donde se descartan las primeras 500 observaciones y posteriormente se toma cada segunda observación. Estas especificaciones se resumen en la siguiente tabla que se presentará antes de cada modelo.

Parámetros		Parámetro Sim.
$M = 2$	$N^* = 4$	$N_{\text{sim}} = 1000$
$J = 2$	$d = 2$	$k^* = 500$
$K = 0$	$n = 350$	$k_{\text{thin}} = 2$

1. Usando las pruebas t clásicas de modelos lineales frequentistas.

Dado que este es un modelo extremadamente sencillo y que se tiene un número pequeño de bases para los polinomios, $N^* = 4$, se presenta por única ocasión, la expansión completa para poderla comparar contra el modelo anterior (2). Esto es[2]:

$$
\begin{aligned}
p_i = P(y_i = 1) = \mathbb{E}[y|\mathbf{x}_i] = \Phi(f(\mathbf{x}_i)) \quad &\Rightarrow \\
\Phi^{-1}(p_i) = \beta_0 + \beta_1 f_1(x_{i,1}) + \beta_2 f_2(x_{i,2}) \qquad &\forall i = 1, n, \ldots, \qquad (3) \\
= \beta_0 \quad & \\
+ \beta_1[w_{1,1} + w_{2,1}x_{i,1} + w_{3,1} + w_{4,1}(x_{i,1} - \tau_{1,1})_+] \qquad &(4) \\
+ \beta_2[w_{1,2} + w_{2,2}x_{i,2} + w_{3,2} + w_{4,2}(x_{i,2} - \tau_{1,2})_+] \qquad &(5)
\end{aligned}
$$

Contrastando (2) contra (3), se puede ver la introducción del componente no lineal a través de las f_j, desglosadas en (4) y (5), para un total de 11 parámetros. Las expansiones truncadas de polinomios son relativamente sencillas y se ve claramente que se les da estructura de recta, permitiendo discontinuidades entre ellas pues $(\cdot)_+$ es una función por partes que se activa si $x_{i,j}$ es mayor que el nodo $\tau\cdot, j$. Aunque esta expansión es aparatosa, el modelo logra hacer excelentes predicciones en cuanto a las regiones. Cabe mencionar que, dado este es un ejemplo introductorio con un número relativamente bajo de observaciones, la estimación de los parámetros, se realizó *dentro de la muestra*[3], esto quiere decir, que el modelo se entrena con las mismas observaciones contra las que se busca predecir[4]. Previo al análisis de convergencia de las cadenas, se hace una exploración preliminar para explicar todos los detalles del modelo. Usando la función de perdida cuadrática, se obtienen los siguientes resultados:

Info. predicción	
Est. Puntual	Media posterior
Precisión	98.85 %
ll	0.03702

$\hat{\beta}$	
$\hat{\beta}_0$	-0.79
$\hat{\beta}_1$	3.35
$\hat{\beta}_2$	0.65

$\hat{\mathbf{w}}$	
-0.52	-1.48
0.39	-0.38
0.09	0.66
1.05	1.32

	$\hat{y} = 0$	$\hat{y} = 1$	
$y = 0$	198	2	200
$y = 1$	2	148	150
	200	150	350

Numéricamente, el modelo se ve bien; la matriz de confusión es idéntica a la del probit anterior y, por ende, la precisión. Sin embargo, se tiene un valor de la función log-loss un poco más bajo, indicando que se tiene un mejor modelo una vez consideradas las probabilidades ajustadas $\hat{\mathbf{p}}$. Sin embargo, a diferencia

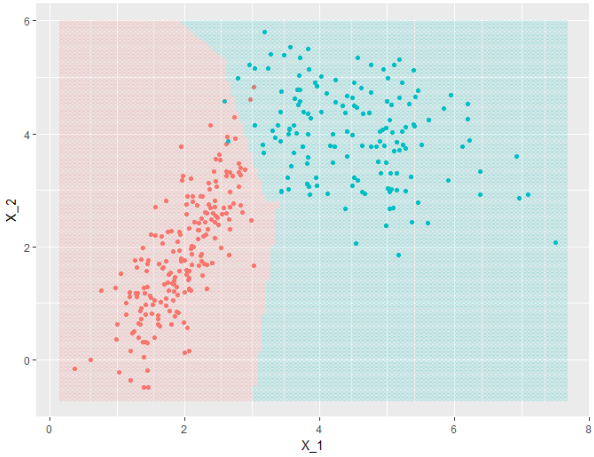
2. Para $w_{l,j}$, se usa la convención de subíndices $l = 1, \ldots, N^*$ de la biyección de la Tabla **??** y $j = 1, \ldots, , d$ para indicar la dimensión

3. *in-sample*

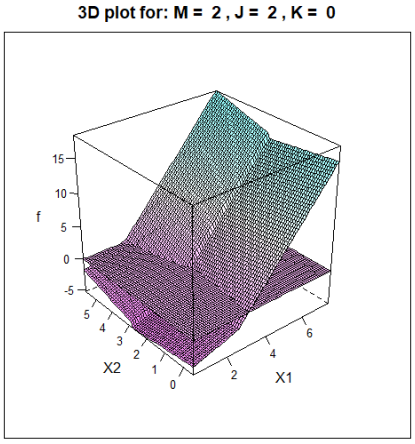
4. El efecto que esto podría tener es que se sobreajuste o se hagan predicciones demasiado acertadas, sin embargo, es normal hacerlo para este tipo de modelos dado n. Además, por lo pronto el objetivo final es dar predicciones a través de las regiones formadas y no tanto para observaciones nuevas. De cualquier forma, se podría hacer separando, antes del análisis, la base de datos en dos, una para entrenar el modelo y otra para probarlo.

del modelo anterior, los parámetros estimados $\hat{\beta}$ y $\hat{\mathbf{w}}$ no se pueden interpretar de la misma manera. En modelos de ML, debido a la complejidad, es mejor tratar a los parámetros como partes funcionales del modelo, que como números con significado. Sin embargo en especial el vector $\hat{\beta}$ puede considerarse como los *pesos* que se le dan a cada transformación no lineal, y su magnitud corresponde a que tanta fuerza tiene esa dimensión en el modelo. De este ejemplo se ve claramente que la primer dimensión, es con la que mejor se están explicando los datos.

La Figura 3a, es clave para entender el modelo, en ella se presentan las dos regiones (de colores) que el modelo detecta para hacer las predicciones. A diferencia de los modelos lineales donde la frontera de



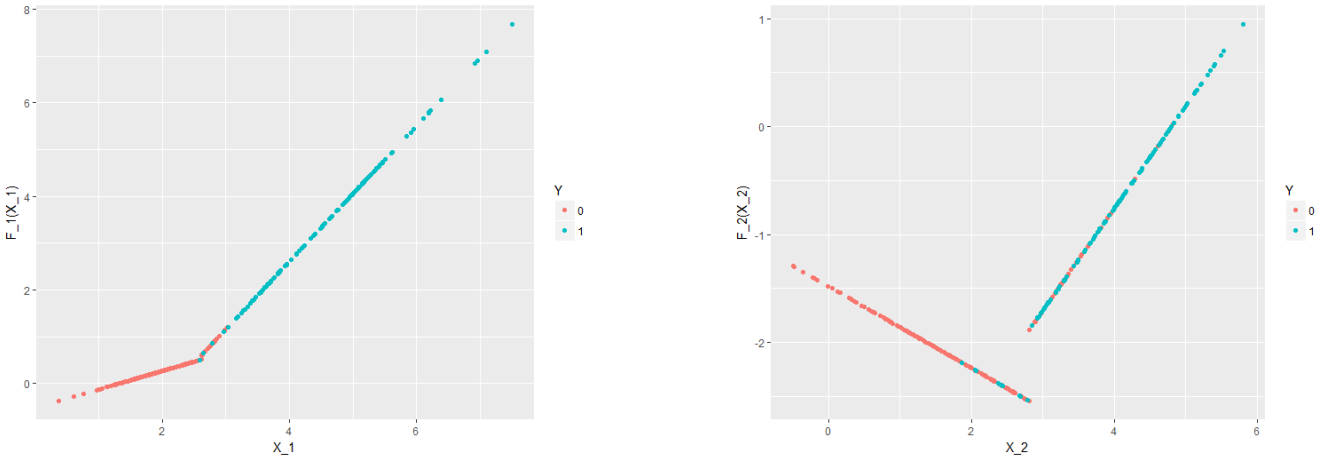
(a) Regiones de predicción para modelo con $M = 2$, $J = 2$ y $K = 0$



(b) Representación en 3D de $\hat{f}$

Figura 3: Visualización de los resultados del modelo

predicción binaria es, por ende, lineal, el modelo presentado en este trabajo permite tener fronteras tan complejas como se quiera (aunque no siempre necesarias). Para este ejemplo en particular, la frontera refleja que se está usando un solo nodo y polinomios lineales discontinuos, es por ello que se da la rugosidad. Encontrar la frontera como tal, resulta una tarea mucho más complicada, pues correspondería a resolver la ecuación $\hat{f}(\mathbf{x}) \equiv 0$; en los GLM, esta frontera es perfectamente lineal y el despeje se puede hacer. Sin embargo, cuando se tienen modelos no lineales, se puede hacer una proyección de ella pues, recordando, lo que interesa es discernir cuando la función $\hat{f}$ sea positiva o negativa. Gracias al hecho que $d = 2$ para este ejemplo, se puede visualizar $\hat{f}(\mathbf{x})$ en la Figura 3b. En esta gráfica, se marca con un plano el *corte* cuando $\hat{f}(\mathbf{x})$ se vuele positiva. Además. se detectan los *pliegues* de discontinuidades derivados del nodo y de la especificación en los parámetros M, J y K. Se hace notar, que esta representación, no es más que la suma ponderada (por $\hat{\beta}$) de las transformaciones no lineales $f_j \quad j = 1, 2$, por lo cual vale la pena visualizarlas en la Figura4.



(a) $\hat{f}_1(x_1)$ (b) $\hat{f}_2(x_2)$

Figura 4: Transformaciones no lineales para cada dimensión.

Aunque la escala vertical de f_j es arbitraria, pues en realidad están ajustando a los residuales parciales, están cumpliendo su propósito de detectar y capturar el patrón que deriva en la separación de los grupos. Para valores izquierdos de ambas variables, se tiene el grupo rojo, para valores mayores, se tiene el grupo azul. Ese efecto, es capturado en el salto que dan las rectas en el nodo, mediante una mayor pendiente en las rectas del lado derecho (mayoritariamente azules), pues, al ser más positivas conforme se avanza en el rango de x, estas tendrán más peso en la función de proyección $\hat{f}$, volviéndola más positiva y por ende, más probable que esa región, contenga observaciones del grupo azul.

Con estas gráficas, se espera dar claridad a todos los componentes del modelo.

0.2.1. Análisis de Sensibilidad

0.2.2. Análisis de Convergencias

0.3. Otros resultados interesantes

Estos ejemplos, son más expositivos que analíticos, es decir, se analizan más las caracteristicas fundamentales que todos los detalles del modelo como se hizo en la sección anterior. Sin embargo, usando el paquete, se puede hacer el análisis de la misma forma.

0.4. Prueba con datos reales