blue!55!white!90!green!90!cyan

Índice

1. Modelos Supervisados **2**

1.1. Introducción: Modelos de Regresión . . . 2
1.2. Modelos lineales continuos . . . 8
1.3. Modelos lineales discretos . . . 14
1.4. Modelos lineales Generalizados . . . 24
1.4.1. Respuestas categóricas . . . 27
1.4.2. Datos de conteo . . . 30
1.4.3. Respuesta Continua . . . 32
1.5. Algoritmos de regularización y estabilidad . . . 32
1.6. Árboles de decisión . . . 34
1.6.1. Árboles de regresión. . . . 49
1.7. K vecinos más cercanos (KNN) . . . 49
1.7.1. Introducción . . . 49
1.7.2. Algoritmo . . . 50
1.7.3. ¿Cuándo usar el algoritmo de KNN? . . . 51
1.8. Máquinas de Soporte Vectorial . . . 52
1.8.1. Introducción . . . 52
1.8.2. Hiperplano y Maximal Margin Classifier . . . 52
1.8.3. Clasificación binaria empleando un hiperplano . . . 53
1.8.4. Support Vector Classifier o Soft Margin SVM . . . 54

1. Modelos Supervisados

1.1. Introducción: Modelos de Regresión

Los problemas de clasificación se predice un 'sí' o un 'no' y después nos fijamos en la probabilidad de equivocarnos, o el número de veces qué nos equivocamos. Sin embargo, las predicciones son de naturaleza probabilística y, en cambio, queremos predecir la probabilidad de un resultado (por ejemplo, la probabilidad de lluvia).

¿Cómo podríamos analizar las predicciones? Por ejemplo, si tenemos dos pronósticos, uno que predice con un 70 % que llueva y el otro predice con una probabilidad del 80 % , ¿cómo podríamos compararlos si llueve? Podemos definir a x como las condiciones climáticas, y a

$$y = \begin{cases} 1 & \text{si llueve} \\ 0 & \text{si no llueve} \end{cases}$$

Deseamos estimar la probabilidad de que llueva dadas las condiciones iniciales, la cual denotaremos por p(x). Observemos que

$$\begin{aligned} p(x) &= P[y = 1|x] \\ &= 1 \cdot P[y = 1|x] + 0 \cdot P[y = 0|x] \\ &= E[y|x]. \end{aligned}$$

Esta última formulación es más general y nos permite predecir, por ejemplo, el número de pulgadas de lluvia que esperamos (una pulgada de lluvia que cae sobre 1 acre de tierra equivale a unos 27 154 galones y pesa unas 113 toneladas).

Este tipo de problemas son llamados de regresión. Estamos tratando de estimar un número de valor real.

En nuestro ejemplo, supongamos que un instituto usa $h_1(x)$ para estimar $p(x)$, mientras que el Servicio Meteorológico utiliza $h_2(x)$. Queremos ver si $h_1(x)$ o $h_2(x)$ predice mejor $p_(x)$, sabiendo que tampoco conocemos $p(x)$.

Por ello, tenemos que confiar solo en los resultados observados y encontrar una manera de "puntuar"las predicciones. Consideraremos una función de "pérdida", que proporciona una medida de cómo puntuar un error. En nuestro primer vistazo, usaremos una función de pérdida cuadrática o cuadrada, que es simplemente el cuadrado de la diferencia entre una probabilidad o expectativa pronosticada y su valor real :

$$L = (h(x) - y)^2$$

Queremos minimizar el valor esperado de nuestra pérdida:

$$E[(h(x) - y)^2]$$

Recordemos que si g es una función y X una v.a discreta, entonces

$$E[g(X)] = \sum_{x \in Dom(X)} P[X = x]g(X)$$

De esta forma, notemos que al minimizar el riesgo, en realidad estamos obligando a h(x) a estar lo más cerca posible de $p(x) = P[y = 1|x]$, pues

$$\begin{aligned} E[(h(x) - y)^2] &= P[y = 1|x](h(x) - y)^2 + P[y = 0|x](h(x) - y)^2 \\ &= p(x)(h(x) - 1)^2 + (1 - p(x))(h(x) - 0)^2 \\ &= p(x)(h(x) - 1)^2 + (1 - p(x))(h(x))^2 \end{aligned}$$

Ahora, para minimizarla, derivamos respecto a h, pues son los valores que deseamos que se aproximen a y e igualamos a cero,

$$\begin{aligned} 0 &= \frac{d}{dh} E[(h(x) - y)^2] \\ &= p(x) \cdot 2(h(x) - 1)(1) + (1 - p(x)) \cdot 2(h(x))(1) \\ &= 2p(x)(h(x) - 1) + 2(1 - p(x))(h(x)) \\ &= 2p(x)h(x) - 2p(x) + 2h(x) - 2p(x)h(x) \\ &= -2p(x) + 2h(x) \\ &= 2[h(x) - p(x)] \end{aligned}$$

La cual se resuelve cuando $h(x) = p(x)$.
Así, el enfoque de esta sección es el aprendizaje por predicción lineal con el enfoque ERM (Empirical risk minimization). Aunque no podemos saber exactamente qué tan bien funcionará un algoritmo en la práctica -el verdadero riesgo- porque no sabemos la verdadera distribución de los datos en los que funcionará el algoritmo, pero podemos medir su desempeño en un conjunto conocido de datos de entrenamiento -el riesgo empírico-.

Se define la clase de funciones afines L_d como

$$L_d = \{h_{w,b} : w \in \mathbb{R}^r, b \in \mathbb{R}\}$$

donde $h_{w,b} : \mathbb{R}^d \to \mathbb{R}$ es tal que :

$$\begin{aligned} h_{w,b}(x) &= \langle w, x \rangle + b \\ &= \sum_{i=1}^{d} w_i x_i + b \end{aligned}$$

De esta manera, L_d es el conjunto de funciones, donde cada una de ellas es parametrizada por $w \in \mathbb{R}^d$ y $b \in \mathbb{R}$, y recibe como input un vector x y regresa un escalar $\langle w, x \rangle + b$. A b se le conoce como el *bias*, el cual se podría interpretar como el error de la aproximación.

Otra forma de denotar el espacio L_d es

$$L_d = \{x \to \langle w, x \rangle + b, w \in \mathbb{R}^d, b \in \mathbb{R}\}$$

A veces es conveniente incorporar a b como la primera coordenada de w , mientras que se añade un 1 en la primera de x. Es decir, se definen w' y x' como

$$w' = (b, w_1, w_2, ..., w_d) \in \mathbb{R}^d \quad \text{y} \quad x' = (1, x_1, x_2, ..., x_d) \in \mathbb{R}^{d+1}$$

tales que

$$h_{w,b}(x) = \langle w, x\rangle + b = \langle w', x'\rangle$$

Así, cada función afín en $\mathbb{R}^d$ puede ser reescrita como una función lineal homogénea en $\mathbb{R}^{d+1}$ bajo la transformación que agrega la constante 1 a la primera entrada de cada vector. De esta manera, para simplificar notación a veces se omitirá el *bias* cuando se hable del conjunto L_d.

La familia de hipótesis de clases de la predicción lineal incluye espacios binarios, predicción por regresión lineal y predicción por regresión logística.

Espacios Binarios (Halfspace)

Consideremos un problema de clasificación en el espacio $\gamma = \{-1, +1\}$ y recordemos a la función signo, la cual denotaremos por $\phi_{sign} : \mathbb{R} \to \gamma$, donde para $a \in \mathbb{R}$,

$$\phi_{sign}(a) = \begin{cases} +1 & \text{si } a \geq 0 \\ -1 & \text{si } a < 0 \end{cases}$$

La clase de espacios binarios o de semiespacios HS_d se define como

$$\begin{aligned} HS_d &= \phi_{sign} \circ L_d \\ &= \{x \to \phi_{sign}(h_{w,b}(x)) : h_{w,b} \in L_d\} \end{aligned}$$

Ejemplo. Para ilustrarlo geométricamente,consideremos el caso $d = 2$ con $w = (w_1, w_2) \in \mathbb{R}^2$, $b \in \mathbb{R}$. En este caso, las hipótesis de clase de espacios binarios HS_d son H_1 y H_2 cuya frontera está delimitada por el hiperplano L_H

$$\begin{aligned} H_1 &:= w_1x_1 + w_2x_2 + b \geq 0 \\ H_2 &:= w_1x_1 + w_2x_2 + b < 0 \\ L_H &:= w_1x_1 + w_2x_2 + b = 0 \end{aligned}$$

Dada L_H, ésta es perpendicular a w , pues para cualesquiera $a, b \in L_H$, $w\dot{(}a - b) = 0$.Las regiones H_1, H_2 intersectan en el punto $\left(0, -\frac{b}{w_2}\right)$ (ordenada al origen de L_H) con el eje x_2. Aquellas $x \in H_1$ serán etiquetadas con +1 y si $x \in H_2$ entonces con -1. Figura 1.
En el contexto de Espacios binarios, un caso realizable o separable es aquel donde es posible separar con un hiperplano todos los ejemplares positivos de los negativos y esto se puede encontrar con el enfoque ERM. Por otro lado, para los casos no separables o agnósticos, es computacionalmente difícil con este enfoque.
[colback=gray!4!white,colframe=blue!40!white!90!green!90!cyan] **Observación 3.1**. Las diferentes clases de hipótesis para la predicción lineal son composiciones de funciones $\phi : \mathbb{R} \to \gamma$ en el espacio L_d. Por ejemplo, para los espacios binarios ϕ es la función signo y $\gamma = \{+1, -1\}$, mientras que para los problemas de regresión ϕ es simplemente la función identidad y $\gamma = \mathbb{R}$.
Programación Lineal para Espacios Binarios

Un problema de programación lineal (LP) pueden ser expresados como la maximización de una función lineal sujeta a inecuaciones lineales, es decir, deseamos encontrar el vector de variables $w \in \mathbb{R}^d$ tal que

$$max_{w\in\mathbb{R}^d} \langle u, v\rangle \tag{1}$$
$$Aw \geq v \tag{2}$$

Donde $A \in M_{m\times d}(\mathbb{R})$ y $v \in \mathbb{R}^m$, $u \in \mathbb{R}^d$ vectores.

Los problema ERM para semiespacios pueden ser expresado como un problema LP. Por simplicidad, consideremos el caso homogéneo. Sea $S = \{(x_i, y_i) : i \in \{1, ..., m\}\}$ el conjunto de entrenamiento de tamaño m. Así el conjunto $\{x_i\}$ es el de vectores de las variables a ingresar y $\{y_i\}$ es el de los escalares a regresar, como estamos en HS_d, entonces los escalares y_i toman valores en el conjunto $\{+1, -1\}$. Considerando un caso realizable, el predictor ERM debe de tener cero errores en el conjunto S. De esta manera, buscamos $w \in \mathbb{R}^d$ tal que para toda $i \in \{1, ..., m\}$,

$$\phi(\langle w, x_i\rangle) = y_i \tag{3}$$

donde ϕ es la función signo.Observemos que $y_i = +1$ si y sólo si $\langle w, x_i\rangle \geq 0$. Del mismo modo, $y_i = -1$ si y sólo si $\langle w, x_i\rangle < 0$. Así, la 3 es equivalente a

$$y_i \langle w, x_i\rangle \ > 0 \tag{4}$$

Sea w^* el vector que satisface 4 y $\overline{w} = \frac{w^*}{\frac{1}{\gamma}}$, donde

$$\gamma = \min_{i\in\{1,...,m\}} y_i \langle w^*, x_i\rangle \ > 0$$

Entonces para toda $i \in \{1, ..., m\}$,

$$y_i \langle w^*, x_i\rangle \geq \gamma$$
$$\Rightarrow \frac{1}{\gamma} y_i \langle w^*, x_i\rangle \geq 1$$

Lo que implica que

$$y_i \langle \overline{w}, x_i\rangle \geq 1 \tag{5}$$

Hemos probado que existe un vector que satisface $y_i \langle w, x_i\rangle \geq$ para toda $i \in \{1, ..., m\}$. Dicho vector es un predictor ERM. Para encontrarlo, usaremos LP. Sea $A \in M_{m\times d}(\mathbb{R})$, tal que para $i \in \{1, ..., m\}$ y $j \in \{1, ..., d\}$, $A_{i,j} = y_i x_{i,j}$, donde $x_{i,j}$ es el j-ésimo elemento del vector $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,d})$.Sea $v = (1, ..., 1) \in \mathbb{R}^m$. Entonces la 5 puede ser reescrita como

$$Aw \geq v \tag{6}$$

Se requiere maximizar una ecuación, como todas las w que cumplen con la restricción 6 son candidatas iguales como hipótesis resultante, se propone la ecuación 1 con $u = (0, ..., 0) \in \mathbb{R}^d$.

Perceptrón para Espacios Binarios

Una implementación con enfoque ERM distinta está dada por el algoritmo llamado Perceptrón (Rosenblatt 1958). Este algoritmo iterativo construye una secuencia de vectores

$w^{(1)}, w^{(2)}, ...$ Se inicializa con $w^{(1)} = (0, ...,) \in \mathbb{R}^d$ y empieza a iterar. Para la t-ésima iteración, verifica si existe $i \in \{1, ..., m\}$ tal que cumpla la condición

$$y_i \left\langle w^t, x_i \right\rangle \leq 0$$

(observar que es la negación de 5) , si es así, entonces se define

$$w^{(t+1)} = w^{(t)} + y_i x_i$$

y procede a realizar la iteración $t+1$, sino se satisface la condición, entonces el vector $w^{(t)}$ es el que buscamos y termina el algoritmo.

El *Teorema 3.1* nos afirma que si tenemos un caso realizable, entonces el algoritmo Perceptrón clasifica correctamente los puntos y existe un número máximo de iteraciones en el mismo.

[colback=gray!4!white,colframe=yellow!40!white!86!orange] ***Teorema 3.1***.

Supongamos que $S = \{(x_i, y_i) : i\{1, ..., m\}\}$ es un conjunto separable (caso realizable). Sea

$$B = min\{||w|| : \forall i \in \{1, ..., m\}, y_i \left\langle w, x_i \right\rangle \geq 1\}$$

y

$$R = \text{máx}\{||x_i|| : i \in \{1, ..., m\}\}$$

El algorimto del Perceptrón se detiene a lo más en $(RB)^2$ iteraciones y si se detiene en la t-ésima iteración, entonces para toda $i \in \{1, ..., m\}$,

$$y_i \left\langle w^{(t)}, x_i \right\rangle > 0$$

Demostración.

Por la definición de la condición de paro del algoritmo del Perceptrón, pues si se detiene en la t-ésima iteración, implica que $\forall i \in \{1, ..., m\}$, $y_i \left\langle w^{(t)}, x_i \right\rangle > 0$, es decir, cumple con la ecuación 5, por lo tanto se ha clasificado correctamente.

Probemos que si el algoritmo ha iterado T veces, entonces $T \leq (RB)^2$, es decir, que el algoritmo se ejecuta a lo más $(RB)^2$ veces.

Sea $w^* \in B$, sabemos que el conjunto B es no vacío por 1.

Dado que $w^{(1)} = (0, ..., 0)$, entonces $\left\langle w^*, w^{(1)} \right\rangle = 0$. Para la t-ésima iteración con $t < T$, supongamos que se actualizó $w^{(t+1)}$ con la i-ésima muestra (x_i, y_i) donde $i \in \{1, ..., m\}$. De esta forma, $y_i \left\langle w^{(t)}, x_i \right\rangle \leq 0$ y se actualizó $w^{(t+1)} = w^{(t)} + y_i x_i$. Por lo tanto

$$\begin{aligned}
\left\langle w^*, w^{(t+1)} \right\rangle - \left\langle w^*, w^{(t+1)} \right\rangle &= \left\langle w^*, w^{(t+1)} - w^{(t)} \right\rangle \\
&= \left\langle w^*, w^{(t)} + y_i x_i - w^{(t)} \right\rangle \\
&= \left\langle w^*, y_i x_i \right\rangle \\
&= y_i \left\langle w^*, x_i \right\rangle \\
&\geq 1 \text{ por 1}
\end{aligned}$$

Lo que implica que

$$\left\langle w^*, w^{(T+1)} \right\rangle = \sum_{t=1}^{T} \left(\left\langle w^*, w^{(t+1)} \right\rangle - \left\langle w^*, w^{(t)} \right\rangle \right) \geq \sum_{t=1}^{T} 1 = T \tag{7}$$

Además, para cada $t < T$, se tiene

$$\begin{aligned} ||w^{(t+1)}||^2 &= ||w^{(t)} + y_i x_i||^2 \\ &= ||w^{(t)}||^2 + 2y_i \left\langle w^{(t)}, x_i \right\rangle + y_i^2 ||x_i||^2 \end{aligned}$$

Debido a que $||x_i|| \leq R$ por la definición de R, implica que $||x_i||^2 \leq R^2$. Por otro lado, sin importar que $y_i = 1$ o $y_i = -1$, ocurre que $y_i^2 = 1$. Además se sabe que $y_i \left\langle w^{(t)}, x_i \right\rangle \leq 0$. De esta forma,

$$||w^{(t+1)}||^2 \leq ||w^{(t)}|| + 2 \cdot 0 + 1 \cdot R^2 = ||w^{(t)}|| + R^2$$

Ahora, probemos que para cualquier iteración t, ocurre que $||w^{(t+1)}||^2 \leq tR^2$.

i) Para $t = 1$. Como $||w(1)|| = 0$ y $||w^{(2)}||^2 \leq ||w^{(1)}||^2 + R^2$, entonces

$$||w^{(2)}||^2 \leq 0^2 + R^2 = R^2$$

ii) Supongamos que para $t \in \mathbb{N}$ se cumple que

$$||w^{(t+1)}||^2 \leq tR^2$$

iii) Probemos para $t + 1$. Sabemos que $||w^{((t+1)+1)}||^2 \leq ||w^{(t+1)}|| + R^2$. Por hipótesis,

$$||w^{((t+1)+1)}||^2 \leq tR^2 + R^2 = (t+1)R^2$$

En particular, se cumple para T, por lo tanto

$$||w^{(T+1)}||^2 \leq TR^2 \quad \Rightarrow \quad ||w^{(T+1)}|| \leq \sqrt{T}R \tag{8}$$

Por definición de B, $||w^*|| = B \geq 0$. Usando 8,

$$||w^*||||w^{(T+1)}|| \leq B\sqrt{T}R \quad \Rightarrow \quad \frac{1}{||w^*||||w^{(T+1)}||} \geq \frac{1}{B\sqrt{T}R}$$

Con lo anterior y la ecuación 7, se obtiene que

$$\frac{\left\langle w^{(T+1)}, w^* \right\rangle}{||w^*||\,||w^{(T+1)}||} \geq \frac{T}{B\sqrt{T}R} = \frac{\sqrt{T}}{BR}$$

En otras palabras, se acaba de probar que el coseno del ángulo entre w^* y $w^{(T+1)}$ es al menos $\frac{\sqrt{T}}{RB}$. Por la desigualdad de Cauchy-Schwartz,

$$\left\langle w^{(T+1)}, w^* \right\rangle \leq ||w^*||\,||w^{(T+1)}|| \quad \Rightarrow \quad \frac{||w^*||\,||w^{(T+1)}||}{\left\langle w^{(T+1)}, w^* \right\rangle} \geq 1$$

Por lo tanto, $1 \geq \frac{\sqrt{T}}{BR}$. De esta manera, concluimos que

$$T \leq (RB)^2$$

[colback=gray!4!white,colframe=blue!40!white!90!green!90!cyan] **Nota**. El algoritmo Perceptrón es seniclo de implementar y garantiza convergencia, sin embargo, depende del valor del parámetro B, el cual puede llegar a tomar exponencialmente grandes en $\mathbb{R}^d$. En tales caso, sería mejor implementar el problema ERM como una solución de programación lineal.

Dimensión VC de los Espacios Binarios

falta ...

1.2. Modelos lineales continuos

La regresión lineal es una herramienta estadística común para modelar la relación entre variables explicativas y se respuesta (valores reales). El dominio es un subconjunto χ de $\mathbb{R}^d$ y el conjunto de respuesta o etiquetas es $Y = \mathbb{R}$.

Nos gustaría obtener una función lineal $h : \mathbb{R}^d \to \mathbb{R}$ que mejor aproxime la relación entre nuestras variables. La Figura 2 muestra un ejemplo de un predictor de regresión lineal para d = 1.

La clase de hipótesis para la predicción por regresión lineal es el conjunto de funciones lineales

$$H_{reg} = L_d = \{x \to \langle w, x\rangle + b : w \in \mathbb{R}^d, b \in \mathbb{R}\}$$

Definimos una función de pérdida para la regresión (*loss function*) denotada por $l(h, (x, y))$.
En clasificación, $l(h, (x, y))$ simplemente indica si $h(x)$ etiqueta correctamente o no, en cambio, en regresión, si deseamos predecir el peso de un bebé de 3kg y obtenemos predicciones de 3.00001 kg y 4 kg, ambas son incorrectas, pero claramente preferiríamos el primero sobre el segundo.
Por ello, necesitamos definir cuánto se penalizará por la discrepancia entre $h(x)$ e y. Una forma común es usar la función de pérdida cuadrática ($squared - lossfunction$), dada por

$$l(h, (x, y)) = (h(x) - y)^2$$

Para esta función de pérdida, la función de riesgo empírica (EMR) se denomina error cuadrático medio ($MeanSquaredError$), dada por

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m}(h(x_i) - yi)^2.$$

Hay una variedad de funciones de pérdida que se pueden utilizar, por ejemplo, la función de pérdida de valor absoluto,

$$l(h, (x, y)) = |h(x) - y|.$$

. La regla ERM para la función de pérdida de valor absoluto se puede implementar mediante programación lineal.
Dado que la regresión lineal no es una tarea de predicción binaria, no podemos analizar su complejidad de muestra utilizando la dimensión VC. Un posible análisis de la complejidad es confiar en el "truco de la discretización" Observación 4.1 en el Capítulo 4). PREGUNTAR .

Mínimos cuadrados.

Los mínimos cuadrados (*Least squares*) es el algoritmo que resuelve el problema ERM para la clase H_{reg} con respecto a la función de pérdida cuadrática. Dado un conjunto de entrenamiento S y usando la versión homogénea de L_d, se enfoca en encontrar w que minimice la expresión

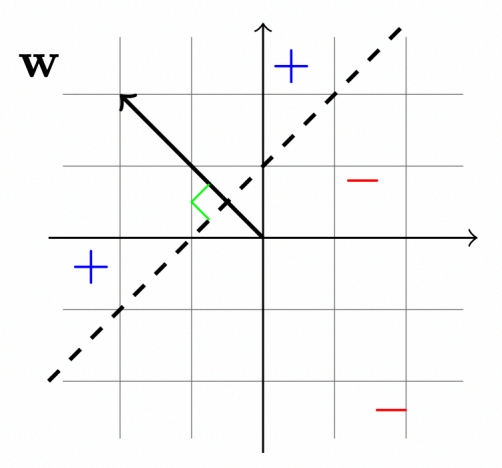

Figura 1: Ejemplo para $\mathbb{R}^2$

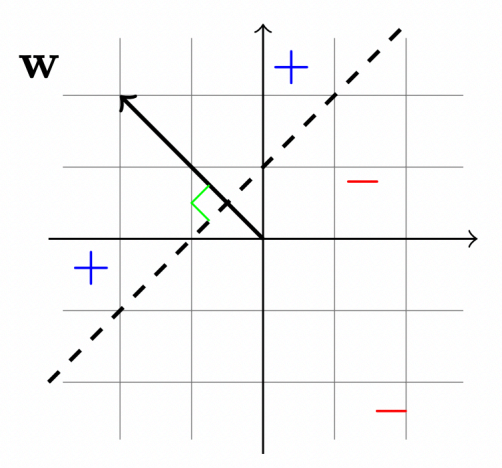

Figura 2: Ejemplo de regresión lineal para $d = 1$

$$L_S(h_w) = \frac{1}{m}\sum_{i=1}^{m}(\langle w, x_i\rangle - y_i)^2$$

Para resolverlo, se calcula el gradiente de la función objetivo y se iguala a cero, es decir,

$$\frac{2}{m}\sum_{i=1}^{m}(\langle w, x_i\rangle - y_i)x_i = 0$$

Lo cual es equivalente a

$$\begin{aligned}
&\sum_{i=1}^{m}(\langle w, x_i\rangle - y_i)x_i = 0 \\
\Rightarrow &\sum_{i=1}^{m}(\langle w, x_i\rangle\, x_i - y_i x_i) = 0 \\
\Rightarrow &\sum_{i=1}^{m}\langle w, x_i\rangle\, x_i - \sum_{i=1}^{m} y_i x_i = 0 \\
\Rightarrow &\sum_{i=1}^{m}\langle w, x_i\rangle\, x_i = \sum_{i=1}^{m} y_i x_i
\end{aligned}$$

Notemos que podemos reescribir el problema como

$$Aw = b$$

donde

$$A = \sum_{i=1}^{m} x_i x_i^T \quad \text{y} \quad b = \sum_{i=1}^{m} y_i x_i$$

$$\begin{aligned}
A &= \begin{pmatrix} x_1 & x_2 & \dots & x_m \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 & \dots & x_m \end{pmatrix}^T \\
&= \begin{pmatrix} x_1 & x_2 & \dots & x_m \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \text{ (Ver Observación 3.3)}
\end{aligned}$$

$$\text{y } b = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

[colback=gray!4!white,colframe=blue!40!white!90!green!90!cyan] **Observación 3.3**. La matriz A es de $p \times m$, donde cada columna i corresponde al vector de variables x_i con $i \in \{1, 2, .., m\}$, además se sabe que x_i tiene p entradas que es el número de variables que determinan la etiqueta de la i-ésima observación. La notación utilizada hace referencia a producto vectorial. Por ejemplo, si $m = 3$ (hubo 3 observaciones) y $p = 2$ (hay dos variables que determinan la etiqueta), entonces

$$
\begin{aligned}
A &= \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}^T \\
&= \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \\
&= x_1 \cdot x_1^T + x_2 \cdot x_2^T + x_3 \cdot x_3^T \\
&= \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} \cdot \begin{pmatrix} x_{11} & x_{12} \end{pmatrix} + \begin{pmatrix} x_{21} \\ x_{22} \end{pmatrix} \cdot \begin{pmatrix} x_{21} & x_{22} \end{pmatrix} + \begin{pmatrix} x_{31} \\ x_{32} \end{pmatrix} \cdot \begin{pmatrix} x_{31} & x_{32} \end{pmatrix} \\
&= x_{11}^2 + x_{12}^2 + x_{21}^2 + x_{22}^2 + x_{31}^2 + x_{32}^2
\end{aligned}
$$

Si A es invertible, la solución está dada por

$$w = A^{-1}b$$

En caso de que A no es invertible, se puede demostrar fácilmente que si las instancias de entrenamiento no generan el espacio $\mathbb{R}^d$, entonces A es no invertible. Sin embargo, podemos encontrar una solución al sistema $Aw = b$ porque b está en el rango de A. Como A es simétrtica, se puede utilizar sus descomposición en eigenvalores dada por

$$A = VDV^T$$

donde D es una matriz diagonal y V es una matriz ortonormal, es decir $V^T \times V = I$.

Se define D^+ como la matriz diagonal tal que

$$D_{i,i}^+ = \begin{cases} 0 & \text{si } D_{i,i} = 0 \\ \frac{1}{D_{i,i}} & \text{si } D_{i,i} \neq 0 \end{cases}$$

De esta forma, se definen

$$A+ = VD^+V^T \quad \text{y} \quad \hat{w} = A^+b$$

Sea v_i la i-ésima columna de la matriz V. Entonces

$$
\begin{aligned}
A\hat{w} &= AA^+b \\
&= (VDV^T)(VD^+V^T)b \\
&= (VD)(V^TV)(D^+V^T)b \\
&= (VD)I(D^+V^T)b \\
&= (VD)(D^+V^T)b \\
&= \sum_{i \in 1,..,m : D_{i,i} \neq 0} v_i v_i^T b
\end{aligned}
$$

Esto imlpica que $A\hat{w}$ es la poryección de b sobre el conjunto generado por los vectores v_i donde $D_{i,i} \neq 0$. Como el generado por $x_1, ..., x_m$ es el mismo que el de los v_i, y b es generado por los x_i, entonces obtenemos que $A\hat{w} = b$.

Regresión lineal para tareas de regresión polinomial .

Algunas tareas de aprendizaje requieren predictores no lineales, como los predictores polinómicos, como en la Figura 3 , en la cual el conjunto de entrenamiento se ajusta mejor usado un predictor polinomial de tercer grado que usando una función lineal.

En esta sección nos enfocaremos en la clase de polinomios de una dimensión de grado n, esto es

$$H^n_{poly} := \{x \to p(x) = a_0 + a_1 x + a_2 x_2 + \ + a_n x_n\}$$

donde $(a_0, ..., a_n)$ es un vector de coeficientes de tamaño $n+1$. Observemos que el conjunto dominio es $\chi = \mathbb{R}$ porque es un polinomio unidimensional y $Y = \mathbb{R}$ conjunto de respuesta. Una forma de resolverlo es por medio de reducción del problema que se ha visto en precios capítulos Para mover el problema de regresión polinomial a un problema de regresión lineal, definimos la función $\phi : \mathbb{R} \to \mathbb{R}^{n+1}$ tal que para $x \in \mathbb{R}$,

$$\phi(x) = (1, x, x^2, ..., x^n)$$

. De esta manera, $p(\phi(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n = \langle a, \phi(x) \rangle$ y se puede encontrar el vector óptimo de coeficientes a utilizando el algoritmo de Mínimos cuadrados.

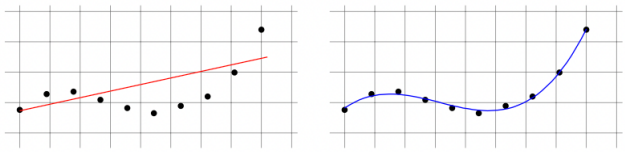

Figura 3: Regresión lineal vs Regresión polinomial

1.3. Modelos lineales discretos

Introducción.

El problema de clasificación en dos grupos puede abordarse introduciendo una variable ficticia binaria para representar la pertenencia de una observación a uno de los dos grupos. Por ejemplo, si se desea discriminar entre créditos que se devuelven o que presentan problemas para su cobro, puede añadirse a la base de datos una nueva variable y que tome el valor 0, cuando el crédito se devuelve sin problemas Y valor 1 en otro caso. El problema de discriminación es equivalente a la previsión del valor de la variable ficticia Y. Si el valor previsto está más próximo a 0 que a 1, clasificaremos al elemento en la primera población. En otro caso, lo haremos en la segunda. Supongamos que Y viene explicada por un conjunto de variables m variables $X_1, X_2, \ldots, X_m$. Por otra parte, podemos pensar en utilizar un modelo de regresión lineal múltiple para explicar el comportamiento de la variable Y, es decir:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m + e,$$

[colback=gray!4!white,colframe=blue!40!white!90!green!90!cyan] **Observación**. La variable del error e viene siendo el bias y $\beta_0, ..., \beta_m$ la $w \in \mathbb{R}^m$ que deseamos estimar utilizando la notación del espacio L_d.
Bajo el supuesto habitual de que $E(e) = 0$, y suponiendo conocidos los valores que toman las variables explicativas (observaciones), tendremos que:

$$E[Y|X_1, \ldots, X_m] = \beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m$$

Observemos que por ser Y una variable binaria (i.e.: sólo podrá tomar los valores 0 y 1), siempre se cumplirá que:

$$E(Y) = 0 \cdot P(Y = 0) + 1 \cdot P(Y = 1) = P(Y = 1), \text{ entonces}$$

$$E(Y|X_1, \ldots, X_m) = P(Y = 1|X_1, \ldots, X_m) = \beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m = Y - e,$$

para $X_1, \ldots, X_m$ conocidos. Observar que esta expresión nos viene a decir que podemos expresar la variable dependiente binaria Y como la probabilidad de "éxito"más un término de perturbación, es decir:

$$Y = P(Y = 1|X_1, \ldots, X_m) + e = E[Y|X_1, \ldots, X_m] + e.$$

Sin embargo, este modelo inicial no será válido para explicar el comportamiento de variables dependientes binarias, pues presenta varios problemas:

1. El error e ya no será una variable aleatoria continua (como ocurría en el los modelos lineales de regresión múltiple (MLRM)), sino que será una variable aleatoria discreta, puesto que, conocidos los valores de las variables explicativas, e sólo puede tomar dos valores determinados. Por tanto, e ya no se distribuirá de forma normal (uno de los supuestos básicos del MRLM). Si bien este supuesto no resulta estrictamente necesario para aplicar mínimos cuadrados ordinarios (MCO), sí es fundamental a la hora de realizar cualquier tipo de inferencia posterior sobre el modelo (intervalos de confianza para los parámetros estimados, contrastes de hipótesis, etc.).

2. El término de perturbación no cumple la hipótesis de homoscedasticidad (la varianza de dicho término no es constante). Esto se da por que $Var(Y|X_1, \ldots, X_m) = P(Y = 1|X_1, \ldots, X_m) * [1 - P(Y = 1|X_1, \ldots, X_m)]$ varía para cada observación i de la muestra. Debido a este problema, MCO no serán eficientes, por lo que resultará necesario recurrir a la estimación por mínimos cuadrados generalizados (MCG).

3. Se corre el riesgo de predecir valores de Y menores que 0 y mayores que 1.

4. Finalmente, la expresión $P(Y = 1|X_1, \ldots, X_m) = \beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m$ nos dice que la probabilidad del suceso de "éxito"viene determinada por una combinación lineal de variables. De ello se deduce que la variación en $P(Y = 1|X_1, \ldots, X_m)$ causada por cambios en alguna de las variables explicativas es constante (y, por tanto, independiente del valor actual de dicha variable explicativa), lo cual es una hipótesis muy poco realista, pues si se perturba la variable X_i, digamos X_i', entonces la variación viene dada por :

$$\begin{aligned}
\frac{\delta P(Y = 1|X_1, \ldots, X_m)}{\delta X_i} &= \frac{P(Y = 1|X_1, \ldots X_i, \ldots, X_m) - P(Y = 1|X_1, \ldots X_i', \ldots, X_m)}{X_i - X_i'} \\
&= \frac{[\beta_0 + \ldots + \beta_i X_i + \ldots + \beta_m X_m] - [\beta_0 + \ldots + \beta_i X_i' + \ldots + \beta_m X_m]}{X_i - X_i'} \\
&= \frac{\beta_i [X_i - X_i']}{X_i - X_i'} \\
&= \beta_i
\end{aligned}$$

Para evitar las inconsistencias anteriores se han desarrollado modelos no lineales, los cuales tratan de resolver los problemas anteriores. La idea consiste utilizar un modelo de la forma

$$Y = f(\beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m) + e,$$

donde f es la función real que depende de la expresión lineal $\beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m$. Con el nuevo modelo, y razonando de forma similar al caso del modelo lineal, se cumplirá:

$$E[Y|X_1, \ldots, X_m] = P(Y = 1|X_1, \ldots, X_m) = f(\beta_1 + \beta_1 X_1 + \ldots + \beta_m X_m).$$

Ahora bien, ¿qué tipo de función f estamos buscando?: obviamente, f deberá ser distinta de la función identidad (para evitar los problemas 3 y 4). La clase de funciones no decrecientes, acotadas entre cero y uno, es la clase de las funciones de distribución, por lo que el problema se resuelve tomando como f cualquier.

Regresión Logística.

La regresión logística se utiliza para tareas de clasificación. Trabajaremos en la familia de funciones h que mapean $\mathbb{R}^d$ en el intervalo $[0, 1]$.

Se puede interpretar entonces $h(x)$ como la probabilidad de que la etiqueta de x sea 1.Así, la hipótesis asociada con la regresión logística H_{sig} es la composición de la función sigmoid $\phi_{sig} : \mathbb{R} \to [0, 1]$ sobre la clase de funciones L_d, es decir,

$$H_{sig} = \phi_{sig} \circ L_d = \{x \to \phi_{sig}(\langle w, x \rangle) : w \in \mathbb{R}^d\}.$$

La función sigmoid usada en regresión logística es la función logic o logística, definida como

$$\phi_{sig}(z) = \frac{1}{1 + exp(-z)}.$$

Notemos que cuando $\langle w, x \rangle$ es muy grande, entonces $\phi_{sig}(\langle w, x \rangle)$ es cercano a 1, mientras que para valores muy pequeños se aproxima a 0 (Figura 4).

[colback=gray!4!white,colframe=blue!40!white!90!green!90!cyan] **Observación.** Recordemos que en el Espacio Binario (Halfspace) (Sección 1) la predicción correspondiente a w es $\phi_{sign}(\langle w, x\rangle)$, por lo tanto las predicciones en regresión logística respecto al espacio binario son muy similares cuando $|\langle w, x\rangle|$.

Cuanto $|\langle w, x\rangle|$ es cercano a 0, $\phi_{sig}(\langle w, x\rangle) \approx \frac{1}{2}$.
En este caso, la regresión logística es incierta respecto a la etiqueta así que determina que es $\phi_{sign}(\langle w, x\rangle)$ con una probabilidad un poco mayor al 50 %. En cambio, el espacio binario da una predicción determinista, 1 o -1 incluso si $|\langle w, x\rangle|$ toma un valor cercano a cero.
Definiremos una función de perdida, es decir, que tan mal predice $h_w(x) \in [0, 1]$ con $h_w \in H_{sig}$dada su verdadera etiqueta $y \in \{-1, +1\}$. De esta manera, deseamos que $h_w(x)$ (probabilidad de que la etiqueta sea 1) tome valores grandes cuando $y = 1$ y que $1-h_w(x)$ (probabilidad de que la etiqueta sea -1) sea grande si $y = -1$. Notemos que

$$\begin{aligned} 1 - h_w(x) &= 1 - \phi_{sig}(\langle w, x\rangle) \\ &= 1 - \frac{1}{1 + exp(-\langle w, x\rangle)} \\ &= \frac{exp(-\langle w, x\rangle)}{1 + exp(-\langle w, x\rangle)} \\ &= \frac{1}{exp(\langle w, x\rangle)} \cdot \frac{1}{1 + exp(-\langle w, x\rangle)} \\ &= \frac{1}{1 + exp(\langle w, x\rangle)} \end{aligned}$$

Por lo tanto, deseamos que cualquier función de perdida crezca monótonamente con

$$\frac{1}{1 + exp(y\,\langle w, x\rangle)}$$

, lo cuál es equivalente a que crezca monótonamente con $1 + exp(-y\,\langle w, x\rangle$.
Si lo anterior se cumple, entonces :

i) La función de pérdida aumentaría cuando $1 - h_w(x)$ (proba de que $y = -1$) es grande y se le asignara la etiqueta $y = 1$, pues $1 + exp(-y\,\langle w, x\rangle)$ sería tal que

$$\begin{aligned} 1 + exp(-y\,\langle w, x\rangle) &= \frac{1}{1 + exp(y\,\langle w, x\rangle)} \\ &= \frac{1}{1 + exp(1 \cdot \langle w, x\rangle)} \\ &= \frac{1}{1 + exp(\langle w, x\rangle)} \\ &= 1 - h_w(x) \text{ que toma un valor grande} \end{aligned}$$

de esta manera, como $1 - h_w(x)$ es grande, entonces es más probable que la etiqueta real fuera $y = -1$, si se le hubiese asignado $y = +11$ (como se desarrolló en la ecuación), al ser la función de pérdida monótonamente creciente con $1 + exp(-y\,\langle w, x\rangle = 1 - h_w(x)$, entonces la función de pérdida tomaría un valor grande (que es lo que deseamos, pues se le estaría asignando una etiqueta incorrecta en este caso).

ii) Del mismo modo, la función de pérdida aumentaría cuando $h_w(x)$ (proba de que $y = +1$) es grande y se le asignara la etiqueta $y = -1$, pues $1 + exp(-y \langle w, x \rangle)$ sería tal que

$$\begin{aligned} 1 + exp(-y \langle w, x \rangle) &= \frac{1}{1 + exp(y \langle w, x \rangle)} \\ &= \frac{1}{1 + exp(-1 \cdot \langle w, x \rangle)} \\ &= \frac{1}{1 + exp(- \langle w, x \rangle)} \\ &= h_w(x) \text{ que toma un valor grande} \end{aligned}$$

Como $h_w(x)$ es grande, entonces es más probable que la etiqueta real fuera $y = +1$, si se le hubiese asignado $y = -1$, al ser la función de pérdida monótonamente creciente con $1 + exp(-y \langle w, x \rangle = h_w(x)$, entonces la función de pérdida tomaría un valor grande (que es lo que deseamos, pues se le estaría asignando una etiqueta incorrecta).

Así, en la regresión logística, la función de perdida penaliza h_w basado en el logaritmo natural de $1 + exp(-y \langle w, x \rangle)$, esto es

$$l(h_w, (x, y)) = log(1 + exp(-y \langle w, x \rangle)$$

Finalmente, dado un conjunto de entrenamiento $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, el problema ERM asociado con la regresión logística es w_{sig} donde w_{sig} es tal que para toda $w \in \mathbb{R}^d$,

$$\frac{1}{m} \sum_{i=1}^{m} log(1 + exp(-y_i \langle w_{sig}, x_i \rangle) \leq \frac{1}{m} \sum_{i=1}^{m} log(1 + exp(-y_i \langle w, x_i \rangle)$$

Otro enfoque, es que como acabamos de ver, una posible solución a las inconsistencias que presentaba el modelo de probabilidad lineal para explicar el comportamiento de una variable dependiente binaria es usar un modelo Logit de la forma:

$$Y = f(\beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m) + e,$$

donde f es la función logística, es decir:

$$f(z) = \frac{\exp(z)}{1 + \exp(z)} = \frac{1}{1 + \exp(-z)}.$$

Por lo tanto tenemos que

$$E[Y|X_1, \ldots, X_m] = P(Y = 1|X_1, \ldots, X_m) = \frac{1}{1 + \exp -(\beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m)}.$$

El modelo puede ser linealizado utilizando la simple transformación

$$\ln \Big(\frac{P(Y = 1|X_1, \ldots, X_m)}{1 - P(Y = 1|X_1, \ldots, X_m)} \Big) = \beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m,$$

comúnmente llamada *transformación logit*. De esta ecuación notamos que

$$O = \frac{P(Y=1|X_1,\ldots,X_m)}{1-P(Y=1|X_1,\ldots,X_m)} = \exp(\beta_0+\beta_1X_1+\ldots+\beta_mX_m) = \exp(\beta_0)\cdot\prod_{i=1}^{m}\exp(\beta_j)^{X_j},$$

donde $\exp(\beta_0)$ y $\exp(\beta_j)$ se les llama *odds (0)* o *ratios de probabilidades* , estos valores indican cuánto se modifican las probabilidades por unidad de cambio en las variables X.

Supongamos que consideramos dos elementos que tienen valores iguales en todas las variables menos en una. Sean $(x_{i1},\ldots,x_{ih},\ldots,x_{im})$ el vector de valores para el primer elemento y $(x_{j1},\ldots,x_{jh},\ldots,x_{jm})$ para el segundo, y todas las variables son las mismas en ambos elementos menos en la variable h donde $x_{ih} = x_{jh}+1$. Entonces, el odds ratio para estas dos observaciones es: $\frac{O_i}{O_j} = e^{\beta_h}$ e indica cuánto se modifica el ratio de probabilidades cuando la variable x_j aumenta en una unidad. Se deduce también que un coeficiente β_i cercano a cero, equivalentemente, un odds-ratio cercano a uno significará que cambios en la variable explicativa X_i asociada no tendrán efecto alguno sobre la variable dependiente Y.

Los parámetros $\beta_0,\beta_1,\ldots,\beta_m$ son estimados por máxima verosimilitud. Para una muestra aleatoria $y_1,\ldots,y_n$ de una distribución Bernoulli con distribución $P(y_i=0|X_1,\ldots,X_m) = 1-p_i$ y $P(y_i=1|X_1,\ldots,X_m)=p_i$, la función de verosimilitud es

$$L(\beta_0,\beta_1,\ldots,\beta_m) = f(y_1,\ldots,y_n;\beta_0,\beta_1,\ldots,\beta_m) = \prod_{i=1}^{n} f_i(y_i;\beta_0,\beta_1,\ldots,\beta_m) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}.$$

De lo cual se obtiene que

$$\ln L(\beta_0,\beta_1,\ldots,\beta_m) = \sum_{i=1}^{n} y_i(\beta_0+\sum_{j=1}^{m}\beta_j x_{ij}) - \sum_{i=1}^{n}\ln(1+e^{\beta_0+\sum_{j=1}^{m}\beta_j x_{ij}}). \tag{9}$$

Diferenciando con respecto a $\beta_0,\ldots,\beta_m$ e igualando a cero tenemos que

$$\text{Para } \beta_0 \quad \sum_{i=1}^{n} y_i = \sum_{i=1}^{n}\frac{1}{1+e^{-\hat{\beta}_0-\sum_{j=1}^{m}\hat{\beta}_j x_{ij}}},$$

$$\text{y para toda } \beta_j \text{ con } j=1,\ldots,m \quad \sum_{i=1}^{n} y_i x_{ij} = \sum_{i=1}^{n}\frac{1}{1+e^{-\hat{\beta}_0-\sum_{j=1}^{m}\hat{\beta}_j x_{ij}}}.$$

Estas ecuaciones se resuelven iterativamente para $\hat{\beta} = (\hat{\beta}_0,\hat{\beta}_1,\ldots,\hat{\beta}_m)$, de la misma manera denotemos a $\hat{Y}$ como el vector estimado bajo el modelo logístico usando los estimadores $\hat{\beta}$.

Para medir la potencia de ajuste del vector estimado $\hat{\beta}$ para el modelo logístico se utiliza la *devianza* o también llamada como la *devianza residual o pseudoresiduos*, esta se puede interpretar como la suma de los errores cuadrados en regresión múltiple lineal. La devianza se define como menos dos veces el logaritmo natural de la log-verosimilitud de los valores ajustados.

$$D = -2\ln L(\hat{\beta}_0,\hat{\beta}_1,\ldots,\hat{\beta}_m) = -2\sum_{i=1}^{n} y_i(\hat{\beta}_0+\sum_{j=1}^{m}\hat{\beta}_j x_{ij}) + 2\sum_{i=1}^{n}\ln(1+e^{\hat{\beta}_0+\sum_{j=1}^{m}\hat{\beta}_j x_{ij}}).$$

Entre más pequeña sea ésta mejor es el ajuste. Para medir la significancia (bajo la hipótesis de que $\beta_j = 0$ $(j = 0, 1, \ldots, m)$) de las variables del modelo se utiliza la prueba univariada de Wald, la cual se obtiene de la siguiente manera

$$W_j = \frac{\hat{\beta}_j}{\hat{\text{SE}}(\hat{\beta}_j)} \qquad j = 0, 1, \ldots, m$$

donde W_j se distribuye como una normal estándar, entonces si $\alpha = P(|z| > W_j)$ es menor a un nivel de significancia α' se rechaza la hipótesis nula de que $\beta_j = 0$ $(j = 0, 1, \ldots, m)$, donde α es el p-value.

Para calcular los intervalos de confianza de los estimadores se usa la siguiente expresión

$$\hat{\beta}_j \pm z_{1-\alpha/2}\hat{\text{SE}}(\hat{\beta}_j) \qquad j = 0, 1, \ldots, m.$$

La matriz de var-cov del estimador $\hat{\beta}$ es $\hat{\text{Var}}(\hat{\beta}) = (X^\top V X)^{-1}$, donde

$$X = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \ldots & x_{nm} \end{bmatrix}, \quad \text{y} \quad V = \begin{bmatrix} \hat{y}_1(1-\hat{y}_1) & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & \hat{y}_n(1-\hat{y}_n) \end{bmatrix}.$$

Y podemos identificar a $\hat{SE}(\hat{\beta}_j)$ como la raíz cuadrada del j-ésimo elemento de la diagonal de la matriz $\hat{\text{Var}}(\hat{\beta})$, es decir, $\hat{\text{SE}}(\hat{\beta}_j) = \sqrt{\hat{\text{Var}}(\hat{\beta})_{jj}}$.

La prueba multivariada de Test para la hipótesis nula de que cada uno de los $m + 1$ coeficientes β sea igual a cero es

$$W = \hat{\beta}^\top \left[\hat{\text{Var}}(\hat{\beta})\right]^{-1} \hat{\beta} = \hat{\beta}^\top \left(X^\top V X\right) \hat{\beta},$$

el cual tiene una distribución chi-cuadrada con $m+1$ grados de libertad, donde el $p-value$ se calcula como $p-value = P[\chi_{p+1} \geq W]$, y si este es menor a un nivel de significancia α' entonces se rechaza la hipótesis nula. Si se requiere hacer la prueba para sólo $h < m + 1$ coeficientes únicamente se tienen que eliminar las β que no se requiera probar y la fila y columna respectiva de la matriz $(X^\top V X)$. Para medir la potencia del ajuste del modelo se utilizan comúnmente los estadísticosχ^2 (Estadístico de Pearson) y D (Devianza)

$$\chi^2 = \sum_{j=1}^{n} \frac{\left[y_i - \left(1 + \exp(-[\hat{\beta}_0 + \sum_{j=1}^{m} \hat{\beta}_j x_{ij}])\right)^{-1}\right]^2}{\left(1 + \exp(-[\hat{\beta}_0 + \sum_{j=1}^{m} \hat{\beta}_j x_{ij}])\right)^{-1}\left[1 - \left(1 + \exp(-[\hat{\beta}_0 + \sum_{j=1}^{m} \hat{\beta}_j x_{ij}])\right)^{-1}\right]}$$

$$D = -2\sum_{i=1}^{n} y_i(\hat{\beta}_0 + \sum_{j=1}^{m} \hat{\beta}_j x_{ij}) + 2\sum_{i=1}^{n} \ln(1 + e^{\hat{\beta}_0 + \sum_{j=1}^{m} \hat{\beta}_j x_{ij}}).$$

Si n es suficientemente largo, en ambos grupos, ambos estadísticos se distribuyen aproximadamente como una $\chi^2_{n-(m+1)}$. Por otra parte, si n es grande y n_1 o n_2 $(n_1 + n_2 = n)$ es pequeño, entonces el uso de estos estadísticos para medir la potencia del ajuste puede ser peligroso, por lo que para valores grandes de χ^2 o D no es evidencia suficiente de falta de ajuste.

Entonces para un nivel de tolerancia específico α_T rechazamos la hipótesis de que el modelo no se distribuye como una $\chi^2_{n-(m+1)}$, si $p-value_D = P[\chi_{n-(m+1)} \leq D]$ o $p-value_{\chi^2} = P[\chi_{p+1} \leq \chi^2]$, es menor que α_T, es decir rechazamos si $\{p - value_D, p - value_{\chi^2}\} \leq \alpha_T$.

Bondad de ajute para la Regresión Logística.

Existen distintos métodos y estad´ısticas para probar la bondad de ajuste de los modelos de regresión logística. Entre ellos están las **Tablas de clasificación.** En ellas e calculan las probabilidades ajustadas $\hat{\pi}_i$ y para cada caso i se obtienen las predicciones (o clasificaciones), "´exito" o "fallo" ("positivo" o "negativo"), dependiendo de si $\hat{\pi}_i$ es mayor o menor que cierto umbral.

Con ello se obtiene la tabla de clasificación 1
donde

- VP : = Verdaderos positivos
- FN = falsos negativos
- VN:= verdaderos negativos
- FP:= falsos positivos

La utilidad del modelo se resumen con la medida de la *Sensibilidad* y la *Especificidad*.

La *Sensibilidad* es la frecuencia relativa de predecir un evento como positivo cuando el evento observado es positivo, es decir,

$$Sensibilidad = \frac{VP}{VP + FN}$$

Por otro lado, la *Especificidad* es la frecuencia relativa de predecir un evento como negativo cuando el evento observado es negativo, de esta forma,

$$Especificidad = \frac{VN}{VN + FP}$$

Así, lo ideal sería que ambas medidas fueran cercanas a 1.

También, existen las **Curvas ROC** (Receiver Operating Characteristic) las cuales grafican la sensibilidad y especificidad para cada umbral. Tradicionalmente, en el eje horizontal se grafica $1especificidad$, y en el eje vertical se grafica la *sensibilidad*.

Con esta orientación de los ejes, un valor del eje x cercano a cero (alta especificidad) generalmente implica un valor del eje y bajo (baja sensibilidad), y viceversa. Todas las curvas ROC comienzan en el punto (0, 0), terminan en el punto (1, 1) y son monótonas crecientes. Un modelo que predice adecuadamente resulta en una curva ROC que crece rápidamente a 1: cuanto más cercana esté a la curva a la parte superior izquierda, mejor serán sus predicciones. Generalmente se calcula el área debajo de la curva ROC, y ésta es una medida de la capacidad predictiva del modelo.

Regresión Probit.

Otra posible solución a las inconsistencias que presentaba el modelo de probabilidad lineal - para explicar el comportamiento de una variable dependiente binaria- es usar un modelo Probit (también llamado modelo Normit) de la forma:

$$Y = f(\beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m) + e,$$

h f es la función de distribución de una normal estándar, es decir

$$f(z) = \int_{-\infty}^{z} \frac{\exp(-t^2/2)}{\sqrt{2\pi}} dt.$$

Por lo tanto tenemos que

$$E[Y|X_1, \ldots, X_m] = P(Y = 1|X_1, \ldots, X_m) = \int_{-\infty}^{\beta_0+\beta_1 X_1+\ldots+\beta_m X_m} \frac{\exp(-t^2/2)}{\sqrt{2\pi}} dt.$$

En este tipo de modelos no resulta posible interpretar directamente las estimaciones de los parámetros β, ya que son modelos no lineales. Lo que haremos en la practica es tomar en cuenta el signo de los estimadores. Si el estimador es positivo, significará que incrementos en la variable asociada causan incrementos en $P(Y = 1|X_1, \ldots, X_m)$ (aunque desconocemos la magnitud de los mismos). Por el contrario, si el estimador muestra un signo negativo, ello supondrá que incrementos en la variable asociada causaran disminuciones en $P(Y = 1|X_1, \ldots, X_m)$.

Podemos ver que $P(Y = 1|X_1, \ldots, X_m) = \Phi(X\beta^\top)$ con $X = (1, X_1, \ldots, X_m)_{n\times(m+1)}$, $\beta = (\beta_0, \ldots, \beta_m)$, y $\Phi(\cdot)$ la función de distribución acumulativa de una normal estándar. Los parámetros β son estimados por máxima verosimilitud. Supongamos que tenemos una muestra de n observaciones independientes y_i y x_{ij} $(j = 1, \ldots, m + 1)$, con $x_{i1} = 1$ para todo $i = 1, \ldots, n$, entonces la log-verosimilitud conjunta es

$$\ln L(\beta) = \ln\Big\{\prod_{i=1}^{n} \Phi(x_i\beta^\top)^{y_i}(1-\Phi(x_i\beta^\top))^{1-y_i}\Big\} = \sum_{i=1}^{n}\Big(y_i \ln\Phi(x_i\beta^\top)+(1-y_i)\ln(1-\Phi(x_i\beta^\top))\Big).$$

De igual forma que en el modelo logit, para maximizar esta log-verosimilitud se tiene que derivar con respecto β igualarse a cero y resolver vía optimización, para β_j $(j = 0, \ldots, m)$ tenemos

$$\sum_{i=1}^{n} \frac{y_i\varphi(x_i\beta^\top)}{\Phi(x_i\beta^\top)} = \sum_{i=1}^{n} \frac{(1-y_i)\varphi(x_i\beta^\top)}{1-\Phi(x_i\beta^\top)},$$

donde $\varphi(\cdot)$ es la función de densidad de una normal estándar.

Una vez obtenido los estimadores $\hat{\beta}$, vía optimización, para medir la potencia del ajuste de esta estimación, se utiliza la devianza $-2\ln L(\hat{\beta})$, entre más pequeña sea ésta mejor es el ajuste. Para medir la significancia (bajo la hipótesis de que $\beta_j = 0$ $(j = 0, 1, \ldots, m)$) de las variables del modelo se utiliza la prueba univariada de Wald, la cual se obtiene de la siguiente manera

$$W_j = \frac{\hat{\beta}_j}{\hat{\mathrm{SE}}(\hat{\beta}_j)} \qquad j = 0, 1, \ldots, m$$

donde W_j se distribuye como una normal estándar, entonces si $\alpha = P(|z| > W_j)$ es menor a un nivel de significancia α' se rechaza la hipótesis nula de que $\beta_j = 0$ $(j = 0, 1, \ldots, m)$, donde α es el p-value.
Para calcular los intervalos de confianza de los estimadores se usa la siguiente expresión

$$\hat{\beta}_j \pm z_{1-\alpha/2}\hat{\mathrm{SE}}(\hat{\beta}_j) \qquad j = 0, 1, \ldots, m.$$

La matriz de var-cov del estimador $\hat{\beta}$ es $\hat{\text{Var}}(\hat{\beta}) = (X^\top V X)^{-1}$, donde

$$X = \begin{bmatrix} 1 & x_{11} & \dots & x_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \dots & x_{nm} \end{bmatrix}, \quad \text{y} \quad V = \begin{bmatrix} \frac{\varphi^2(x_1\beta^\top)}{\hat{y}_1(1-\hat{y}_1)} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{\varphi^2(x_n\beta^\top)}{\hat{y}_n(1-\hat{y}_n)} \end{bmatrix},$$

donde $\varphi(z) = \exp(-\frac{z^2}{2})/\sqrt{2\pi}$.Podemos identificar a $\hat{SE}(\hat{\beta}_j)$ como la raíz cuadrada del j-ésimo elemento de la diagonal de la matriz $\hat{\text{Var}}(\hat{\beta})$, es decir, $\hat{\text{SE}}(\hat{\beta}_j) = \sqrt{\hat{\text{Var}}(\hat{\beta})_{jj}}$.La prueba multivariada de Test para la hipótesis nula de que cada uno de los $m+1$ coeficientes β sea igual a cero es

$$W = \hat{\beta}^\top \Big[\hat{\text{Var}}(\hat{\beta})\Big]^{-1} \hat{\beta} = \hat{\beta}^\top \Big(X^\top V X\Big) \hat{\beta},$$

el cual tiene una distribución chi-cuadrada con $m+1$ grados de libertad, donde el $p-value$ se calcula como $p - value = P[\chi_{m+1} \geq W]$, y si este es menor a un nivel de significancia α' entonces se rechaza la hipótesis nula. Si se requiere hacer la prueba para sólo $h < m+1$ coeficientes únicamente se tienen que eliminar las β que no se requiera probar y la fila y columna respectiva de la matriz $(X^\top V X)$.
Para medir la potencia del ajuste del modelo se utilizan comúnmente estos estadísticos

$$\text{Estadístico de Pearson} \quad \chi^2 = \sum_{j=1}^{n} \frac{(y_i - \Phi(x_i\hat{\beta}^\top)^2}{\Phi(x_i\hat{\beta}^\top)(1 - \Phi(x_i\hat{\beta}^\top))} \qquad \text{y}$$

$$\text{Devianza} \quad D = -2\sum_{i=1}^{n} \Big(y_i \ln \Phi(x_i\hat{\beta}^\top) + (1 - y_i) \ln(1 - \Phi(x_i\hat{\beta}^\top))\Big).$$

Si n es suficientemente largo, en ambos grupos, ambos estadísticos se distribuyen aproximadamente como una $\chi^2_{n-(m+1)}$. Por otra parte, si n es grande y n_1 o n_2 $(n_1 + n_2 = n)$ es pequeño, entonces el uso de estos estadísticos para medir la potencia del ajuste puede ser peligroso, por lo que para valores grandes de χ^2 o D no es evidencia suficiente de falta de ajuste.
Entonces para un nivel de tolerancia específico α_T rechazamos la hipótesis de que el modelo no se distribuye como una $\chi^2_{n-(m+1)}$, si $p-value_D = P[\chi_{n-(m+1)} \leq D]$ o $p-value_{\chi^2} = P[\chi_{p+1} \leq \chi^2]$, es menor que α_T, es decir rechazamos si $\{p - value_D, p - value_{\chi^2}\} \leq \alpha_T$.

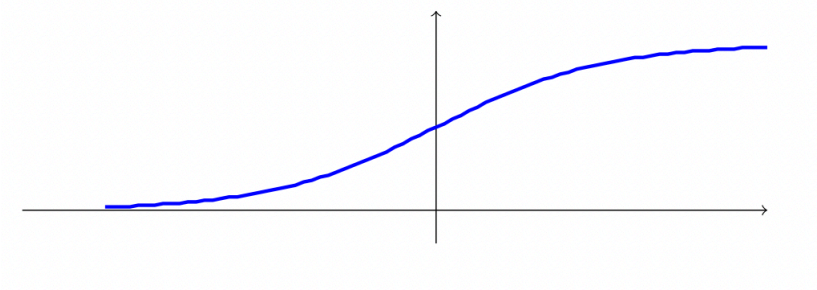

Figura 4: Función logic

	Positivos	**Negativos**
Positivos	VP	FP
Negativos	FN	VN

Cuadro 1: Tabla de clasificación. *Predicciones* (en itálicas) y **Observaciones** (en negritas)

1.4. Modelos lineales Generalizados

Los modelos lineales generalizados (MLG) son aquellos en los que una variable, la cual se conoce como *dependiente* o *respuesta*, se explica por medio de la combinación lineal de otras variables, que son llamadas *independientes*, *explicativas* o como *covariables*, cuando se trata de categorías se le conoce como *factores*.

Los MLG son una extensión de los modelos lineales y se caracterizan por

- *Componente aleatorio*. Un vector de n observaciones, digamos

$$y = (y_1, ..., y_n)$$

, donde y_i es la realización de la variable aleatoria Y_i. El vector de v.a independientes $Y = (Y_1, ..., Y_n)$ es tal que toda Y_i pertenece a la familia exponencial y

$$E[Y_i] = \mu$$

- *Componente sistemática*. El conjunto de covariables $\{x_1, ..., x_p\}$ forma un predictor lineal dado por

$$\eta = \sum_{j=1}^{p} x_i B_j$$

o bien, en notación matricial, sea $X \in M_{n\times p}(\mathbb{R})$, es decir, una matriz de $n \times p$, cuya $j - sima$ columnas vienen dadas por la covariable $x_j = (x_{1j}, ..., x_{nj})$ y $B \in_{p\times 1} M(\mathbb{R})$ el vector de parámetros desconocidos cuyo j-ésimo renglón está dado por β_i, entonces

$$\begin{aligned}\eta &= \begin{pmatrix} x_1 & x_2 & ... & x_p \end{pmatrix} \times \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} \\ &= X\beta\end{aligned}$$

- *Liga o función de enlace*: La función de enlace relaciona los componentes aleatorio y sistemático:

$$\eta_i = g(\mu_i)$$

Common distributions with typical uses and canonical link functions

Distribution	Support of distribution	Typical uses	Link name	Link function, $\mathbf{X}\boldsymbol{\beta} = g(\mu)$	Mean function
Normal	real: $(-\infty, +\infty)$	Linear-response data	Identity	$\mathbf{X}\boldsymbol{\beta} = \mu$	$\mu = \mathbf{X}\boldsymbol{\beta}$
Exponential Gamma	real: $(0, +\infty)$	Exponential-response data, scale parameters	Inverse	$\mathbf{X}\boldsymbol{\beta} = \mu^{-1}$	$\mu = (\mathbf{X}\boldsymbol{\beta})^{-1}$
Inverse Gaussian	real: $(0, +\infty)$		Inverse squared	$\mathbf{X}\boldsymbol{\beta} = \mu^{-2}$	$\mu = (\mathbf{X}\boldsymbol{\beta})^{-1/2}$
Poisson	integer: $0, 1, 2, \ldots$	count of occurrences in fixed amount of time/space	Log	$\mathbf{X}\boldsymbol{\beta} = \ln(\mu)$	$\mu = \exp(\mathbf{X}\boldsymbol{\beta})$
Bernoulli	integer: $\{0, 1\}$	outcome of single yes/no occurrence	Logit	$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})} = \frac{1}{1+\exp(-\mathbf{X}\boldsymbol{\beta})}$
Binomial	integer: $0, 1, \ldots, N$	count of # of "yes" occurrences out of N yes/no occurrences			
Categorical	integer: $[0, K)$	outcome of single K-way occurrence			
	K-vector of integer: $[0, 1]$, where exactly one element in the vector has the value 1				
Multinomial	K-vector of integer: $[0, N]$	count of occurrences of different types (1 .. K) out of N total K-way occurrences			

Adicionalmente para las respuestas binarias se tienen las siguientes funciones de respuesta:

$$g(\mu) = X\beta = \Phi(\mu) \qquad \text{y} \qquad g(\mu) = X\beta = \log\{-\log(1-\mu)\}$$

Que son la probit y la log-log complementaria.

¿Cómo se encuentran los parámetros?

¡Por máxima verosimilitud!, básicamente, se utiliza la función de distribución $f_y(y;\beta)$ con ella se

1. Calcula la función de log-verosimilitud $\log l(;y) = \log \mathcal{L}(;y) = \log\{\Pi_{i=1}^{n} f_y(y_i;\beta_i)\}$
2. Se derivan los parámetros e igualan a cero: $\frac{\partial l(;y)}{\partial \beta} = (\frac{\partial l(;y)}{\partial \beta_1}, \frac{\partial l(;y)}{\partial \beta_2}, \ldots, \frac{\partial l(;y)}{\partial \beta_m})^T = 0$
3. Se despejan los parámetros (si es posible) o se estiman a través d eun algoritmo para encontrar los valores de éstos.

También se puede intentar utilizar mínimos cuadrados ordinarios y mínimos cuadrados parciales, el tema es que la estimación de parámetros se podría complicar para funciones liga que no son identidad.

¿Qué pruebas de hipótesis se aplican?

Para el predictor lineal $X_i^T\beta$ supongamos que se particiona los estimadores en dos $\beta = (\beta_1, \beta_2)$ y se interesa en probar la hipótesis:

$$H_0 : \beta_1 = \beta_{1,0} \qquad \text{vs} H_1 : \beta_1 \neq \beta_{1,0}$$

Donde:
La longitud de β_1 es $q \leq m$ y β_2 es arbitraria.
$\beta_{1,0} = 0$ Dado lo anterior hay dos marcos de hipótesis:

Prueba de Wald:
Sea $\hat{\beta}_{EMV} = (\hat{\beta}_{1,EMV}, \hat{\beta}_{2,EMV})$
Bajo H_0

$$(\hat{\beta}_{1,EMV} - \hat{\beta}_{1,0})^T \hat{V}[\hat{\beta}_{1,EMV}]^{-1}(\hat{\beta}_{1,EMV} - \hat{\beta}_{1,0}) \longrightarrow_d \frac{2}{q}$$

donde $\hat{V}[\hat{\beta}_{1,EMV}]^{-1}$ es la inversa de $q \times q$ submatriz de varianzas de β_1 evaluadas en $\hat{\beta}_{1,EMV}$

Prueba de cociente de verosimilitud:
Obtener el mejor modelo sin restricciones.
Obtener el mejor modelo bajo H_0 descrito anteriormente:
Entonces bajo H_0

$$2(l(\hat{\beta}_{EMV}; y) - l(\hat{\beta}_{0,EMV}; y)) \longrightarrow_d \chi^2_q$$

La familia exponencial y los MLG.

La familia exponencial está estrechamente relacionada con los MLG. Sea Y una va.a con distribución perteneciente a la familia exponencial, entonces su función de densidad está dada por

$$f_Y(y) = c(y, \phi) exp\left(\frac{y\theta - a(\theta)}{\phi}\right)$$

Donde

- θ es el parámetro canónico
- ϕ es el parámetro de dispersión
- $a : \mathbb{R} \to \mathbb{R}$ es una función de θ
- $c : \mathbb{R}^2 \to \mathbb{R}$ es una función de y y ϕ

Algunas de las distribuciones más conocidas como la Binomial, la Geométrica, la Binomial negativa, la Poisson, la Gama, la Normal y la Beta, pertenecen a la familia exponencial.

Ejemplo. La distribución Binomial pertenece a la familia exponencial. Sea $Y \sim Bin(n, \pi)$, con $n \in \mathbb{N}$ y $\pi \in (0, 1)$. Entonces la función de masa de probabilidad está dada por

$$\begin{aligned}
P(Y = y) &= \binom{n}{y} \pi^y (1-\pi)^{n-y} \\
&= \binom{n}{y} \pi^y (1-\pi)^{-y} \cdot \frac{1}{(1-\pi)^{-n}} \\
&= \binom{n}{y} \left(\frac{\pi}{1-\pi}\right)^y \cdot \frac{1}{\frac{1}{(1-\pi)^n}} \\
&= \binom{n}{y} \left(\frac{\pi}{1-\pi}\right)^y \cdot \frac{1}{\left(\frac{1-\pi+\pi}{1-\pi}\right)^n} \\
&= \binom{n}{y} \left(\frac{\pi}{1-\pi}\right)^y \cdot \frac{1}{(1+\frac{\pi}{1-\pi})^n} \\
&= \binom{n}{y} exp\left\{ln\left(\frac{\left(\frac{\pi}{1-\pi}\right)^y}{(1+\frac{\pi}{1-\pi})^n}\right)\right\} \\
&= \binom{n}{y} exp\left\{ln\left(\frac{\pi}{1-\pi}\right)^y - ln\left(1+\frac{\pi}{1-\pi}\right)^n\right\} \\
&= \binom{n}{y} exp\left\{\frac{y \cdot ln\left(\frac{\pi}{1-\pi}\right) - n \cdot ln\left(1+\frac{\pi}{1-\pi}\right)}{1}\right\}
\end{aligned}$$

Se proponen :

$$\theta = ln\left(\frac{\pi}{1-\pi}\right) \quad , \phi = 1, \quad c(y,\phi) = \binom{n}{y}, \quad , \text{ y } a(\theta) = n \cdot ln\left(1 + \frac{\pi}{1-\pi}\right) = n \cdot ln\left(1 + e^{\theta}\right)$$

Por lo tanto, la distribución Binomial pertenece a la familia exponencial. Algunas distribuciones de la familia exponencial son la Binomial, Poisson, Normal, Gamma, Gaussiana inversa y Binomial negativa.

[colback=gray!4!white,colframe=blue!40!white!90!green!90!cyan] **Observación**. El modelo lineal clásico, o lineal normal es uno de los más importantes y utilizados de los MLG. Es crucial para para el estudio de los demás modelos pertenecientes a la clase de MLG. Veremos 3 formas de los MLG: para respuestas categóricas, para datos de conteo y para respuestas continuas.

1.4.1. Respuestas categóricas

Recordemos que las variables categóricas toman los valores de un número posible de categorías. Existen dos tipos de variables categóricas: las variables cuyas categorías tienen un orden natural (ordinal) y las que no lo tienen (nominal).

Para variables binarias, digamos $y = 0$ o $y = 1$. Si π es la probabilidad de que $y = 1$, entonces $y \sim B(1, \pi)$. El MLG Bernoulli (Binomial con n = 1) es

$$Y \sim B(1, \pi), g(\pi) = X\beta$$

Recordemos que la proporción $\frac{\pi}{(1-\pi)}$ se llama odds o momios, e indica proporcionalmente cuanto más probable es la ocurrencia del evento comparada con la no-ocurrencia. Para las respuestas binarias tenemos **la regresión logística** o los **modelos Probit** que se vieron anteriormente.

Datos binarios agrupados.

Cuando todas las variables explicativas son categóricas es posible expresar un conjunto de datos en forma agrupada. Un grupo consiste de todos los casos con los mismos valores de las variables explicativas y puede corresponder a un conjunto de riesgos homogéneo. En el caso de una respuesta binaria, una vez que los datos están agrupados, la respuesta observada es el número de eventos que ocurren en cada grupo. La notación es la siguiente:

- $m :=$ número de grupos.
- $n_i :=$ número de casos en el grupo i
- $y_i :=$ número de eventos ocurridos en el grupo i.
- $\pi_i :=$ probabilidad de que el evento ocurra para un caso en el grupo i
- $n :=$ tamaño de la muestra, $n = \sum_{i=1}^{m} n_i$

Por lo tanto, y_i es el número de ocurrencias del evento, de un máximo de n_i, donde la probabilidad de ocurrencia del evento es π_i.

Por lo tanto, la respuesta observada tiene una distribución, $y_i B(n_i, \pi_i)$, donde la probabilidad π_i se modela como una función de las variables explicativas.

Variable categórica con más de dos categorías.

Ahora bien, si tuviéramos una respuesta categórica con r categorías, para la respuesta se define Para la respuesta se definen $r1$ variables respuesta indicadoras y_j, con $j \in \{1, ..., r1\}$ y definimos

$$y_j = \begin{cases} 1 & \text{si la respuesta está en el nivel } j \\ 0 & \text{en otro caso} \end{cases}$$

. De esta manera, la respuesta

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_{r-1} \end{pmatrix}$$

multivariada. Los modelos con respuesta nominal y ordinal entran en la clase de los MLG multivariados, usando la familia exponencial multivariada.

Dadas n observaciones independientes de la respuesta y, es de interés obtener el número de veces que la categoría j ocurre,

$$n_j = \sum_{i=1}^{n} y_{ij}$$

Sea π_j la probilidad de qie la respuesta pertenezca a la categoría j, entonces

$$\sum_{j=1}^{r} \pi_j = 1, \quad , \text{ y} \quad n = \sum_{j=1}^{r} n_j$$

La distribución conjunta de $n_1, \ldots, n_r$ es multinomial con función de masa de probabilidad dada por

$$\mathbb{P}(n_1, ..., n_r) = \frac{n!}{n_1! \times \ldots \times n_r!} \pi_1^{n_1} \ldots \pi_r^{n_r}$$

Respuesta ordinal.

Sea y una respuesta ordinal con r categorías ordenadas. Se define una variable continua y^* y umbrables θ_j tales que para todo $j \in \{1, ..., r\}$

$$y = j \quad \text{si } \theta_j \leq y^* \leq \theta_j$$

El modelo se define en términos de las probabilidades acumulativas τ_j a las variables explicativas. Suponga que

$$y^* = -X'\beta + \varepsilon \text{ donde} E[\varepsilon] = 0$$

Lo anterior implica que

$$E[y^*] = X'\beta$$

Finalmente,

$$\tau_j = P(\varepsilon \leq \theta_j + X'\beta)$$

donde la distribución de ε determina la forma exacta del modelo.

Modelo logístico acumulado.

En el modelo logístico acumulado o de momios proporcionales se supone que se tiene una distribución logística estándar, dada por :

$$P(\varepsilon \leq y) = \frac{1}{1+e^{-x}}$$

De esta manera ,

$$\begin{aligned}
\tau_j = P(\varepsilon \leq \theta_j + x'\beta) &= \frac{1}{1+e^{-(\theta_j+x'\beta)}} \\
\Rightarrow ln\left(\frac{\tau_j}{1-\tau_j}\right) &= ln\left(\frac{\frac{1}{1+e^{-(\theta_j+x'\beta)}}}{1-\frac{1}{1+e^{-(\theta_j+x'\beta)}}}\right) \\
&= ln\left(\frac{\frac{1}{1+e^{-(\theta_j+x'\beta)}}}{\frac{1+e^{-(\theta_j+x'\beta)}-1}{1+e^{-(\theta_j+x'\beta)}}}\right) \\
&= ln\left(\frac{\frac{1}{1+e^{-(\theta_j+x'\beta)}}}{\frac{e^{-(\theta_j+x'\beta)}}{1+e^{-(\theta_j+x'\beta)}}}\right) \\
&= ln\left(\frac{1}{e^{-(\theta_j+x'\beta)}}\right) \\
&= ln\left(e^{(\theta_j+x'\beta)}\right) \\
&= \theta_j + x'\beta \text{ para } j \in \{1,...,r-1\}
\end{aligned}$$

Donde

- Los θ_j son términos de intersección que dependen de j.
- x no contiene al 1 , es decir, no hay otro término de intersección.
- Los coeficientes β no dependen de j.

Modelo log-log complementario acumulado
En este modelo, la distribución de ε es la de valores extremos mínimos, por lo tanto, el modelo es

$$
\begin{aligned}
ln(-ln(1-\tau_j)) &= ln(-ln(1-\frac{1}{1+e^{-(\theta_j+x'\beta)}})) \\
&= ln(-ln(\frac{1+e^{-(\theta_j+x'\beta)}-1}{1+e^{-(\theta_j+x'\beta)}})) \\
&= ln(-ln(\frac{e^{-(\theta_j+x'\beta)}}{1+e^{-(\theta_j+x'\beta)}})) \\
&= ln(-ln(\frac{e^{-(\theta_j+x'\beta)}}{1+e^{-(\theta_j+x'\beta)}})) \\
&= ln(-ln(\frac{1}{e^{-(\theta_j+x'\beta)}(1+e^{-(\theta_j+x'\beta)})})) \\
&= ln(-ln(\frac{1}{e^{-(\theta_j+x'\beta)}+e^0}) \\
&= ln(ln\left(\frac{1}{e^{-(\theta_j+x'\beta)}+1}\right)^{-1}) \\
&= ln(ln(e^{-(\theta_j+x'\beta)}+1))
\end{aligned}
$$

Respuesta nominal.

Los modelos para respuestas nominales se conocen como regresión nominal, regresión politómica(polytomous), regresión policotómica (polychotomous) o regresión multinomial.

Sea y una respuesta nominal y con r categorías nominales (no ordenadas). Las probabilidades multinomiales son

$$\pi_j = P(y=j) \text{ tales que } \sum_{j=1}^{r}\pi_j = 1$$

Los momios de la categoría j relativos a la categoría base r se modelan como:

$$ln\left(\frac{\pi_j}{\pi_r}\right) = \theta_j + x'\beta_j, \text{ con } j \in \{1,...,r-1\}$$

Se sigue que

$$\pi_r = \frac{1}{1+\sum_{k=1}^{r-1}e^{\theta_k+x'\beta_k}} \quad \text{donde } \pi_j = \pi_r e^{\theta_j+x'\beta_j} \text{ para } j \in \{1,...,r-1\}$$

1.4.2. Datos de conteo

Estudiaremos los MLG cuando la respuesta es una variable de conteo, por ejemplo, número de muertes,número de reclamaciones o número vehículos asegurados, y se deseamos explicarla en términos de otras variables. Veremos dos tipos de regresiones para esta tarea : la regresión Poisson y la Regresión binomial negativa.

Regresión Poison.

Cuando la variable respuesta es un conteo, frecuentemente se usa la distribución Poisson. En la regresión Poisson, la media μ se explica en términos de las variables explicativas xs, usando la función de enlace apropiada. El modelo de regresión Poisson se define como

$$Y \sim P(\mu), g(\mu) = X\beta$$

donde la función de enlace generalmente es $g(\mu) = ln(\mu)$, aunque también se puede usar la identidad $g(\mu) = \mu$, pero a no garantiza que los conteos sean positivos.

Suponga que el modelo tiene una variable explicativa x_1, entonces

$$x = \begin{pmatrix} 1 \\ x_1 \end{pmatrix}, \quad , \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$
$$g(\mu) = x'\beta = \beta_0 + \beta_1 x_1$$

Con la función de enlace ln, el valor esperado de y es

$$\mu = e^{\beta_0 + \beta_1 x_1}$$

[colback=gray!4!white,colframe=blue!40!white!90!green!90!cyan] **Observación**.
Si x_1 aumenta en una unidad entonces, $e^{\beta_0 + b_1(x_1+1)} = e^{\beta_0 + \beta_1 x_1} e^{\beta_1}$
lo que implica que existe un incremento multiplicativo en μ con un valor de e^{β_1}.
Si se tiene una variable explicativa categórica con r niveles donde el nivel r es el base, entonces la variable se reemplaza por $r1$ variables indicadoras:

$$x_1, ..., x_{r1}$$

y el modelo es

$$g(\mu) = \beta 0 + \beta_1 x1 + \ + \beta_{r1} x_{r1}$$

. Con la función de enlace ln, el valor esperado de y es

$$\mu = e^{0 + \beta_1 x_1 + ... + \beta_{r-1} x_{r-1}}$$

Cuando la variable explicativa est´a en el nivel base r se
tiene que $\mu = e^{\beta_0}$ y si se encuentra en el nivel j , $\mu = e^{\beta_0 + \beta_j} = e^{\beta_0} e^{\beta_j}$. Lo que implica que existe un incremento multiplicativo de e^{β_j} respecto al nivel base.

Notemos que

- Si $\beta_j = 0$. La respuesta media para la categoría j es la misma que para el nivel base.
- Si $\beta_j > 0$. Se incrementa ya que $e^{\beta_j} > 1$
- si $\beta_j < 0$. La respuesta media en la categoría j es decreciente debido a que $e^{\beta_1} < 1$

Regresión binomial negativa.

La distribución binomial negativa puede utilizarse para datos de conteo, cuando la sobre-dispersión de los datos está explicada por la heterogeneidad de la media sobre la población. El modelo de regresión binomial negativa, con la función de enlace ln, es

$$Y \sim NB(\mu, k), \quad ln(\mu) = ln(n) + X\beta$$

1.4.3. Respuesta Continua

Cuando las variables respuesta son continuas, no negativas y sesgadas a la derecha existen dos opciones que pueden utilizarse para modelar:

- Usar una transformación para normalidad, y usar el modelo lineal normal con la respuesta transformada.
- Usar MLG con una distribución para la variable respuesta que esté definida en los reales no negativos, por ejemplo la distribución gamma o Gaussiana inversa.

Regresión Gamma

El modelo de regresión gama generalizado está dado por

$$y \sim Gamma(\mu, v), \quad g(\mu) = x'$$

La función de enlace canónica para la distribución gamma es la función inversa. Como los parámetros de un modelo con función de enlace inversa son difíciles de interpretar, es común usar la función de enlace ln.

Regresión Gaussiana inversa

Se define como

$$y \sim IG(\mu, \sigma^2) \quad g(\mu) = x'\beta$$

La funicón de enlace canónica es $g(\mu) = \mu^{-2}$, sin embargo, es común usar la función ln.

1.5. Algoritmos de regularización y estabilidad

El nuevo paradigma de aprendizaje que presentamos en este capítulo se denomina Minimización de pérdida regularizada (Regularized Loss Minimization) o RLM para abreviar. En RLM minimizamos la suma del riesgo empírico L_s con una función de regularización R.

Intuitivamente, la función de regularización mide la complejidad de las hipótesis, es como un estabilizador del algoritmo de aprendizaje. Intuitivamente, un algoritmo se considera estable si un ligero cambio en su entrada no cambia mucho su salida. Definiremos formalmente la noción de estabilidad (lo que entendemos por "cambio leve de entrada" y por "no cambia mucho la salida") y demostraremos su estrecha relación con la capacidad de aprendizaje.

Finalmente, mostraremos que el uso de la norma l_2 al cuadrado como función de regularización estabiliza todos los problemas de aprendizaje.

Minimización de pérdidas regularizadas

La Minimización de Pérdida Regularizada (RLM) es una regla de aprendizaje en la que minimizamos conjuntamente el riesgo empírico y una función de regularización. Formalmente, una función de regularización es un mapeo $R : \mathbb{R}^d \to \mathbb{R}$ y ka función que miniza la perdida regularizada nos da w que minimice la expresión

$$L_s(W) + R(w)$$

La minimización de pérdidas regularizada comparte similitudes con los algoritmos de longitud mínima de descripción y la minimización de riesgos estructurales. Intuitivamente, la

"complejidad" de las hipótesis se mide por el valor de la función de regularización y el algoritmo equilibra el riesgo empírico bajo con hipótesis "más simples" o "menos complejas".

Una de las función de regularización sencilla está dada por

$$R(w) = \lambda||w||^2$$

donde $\lambda > 0$ es un escalar y la norma $||\cdot|| : \mathbb{R}^d \to \mathbb{R}$ es la de l_2 dada por

$$||w|| = \sqrt{\sum_{i=1}^{2} w_i^2}$$

Por lo tanto, al regla de aprendizaje está dada por

$$A(S) = w \text{ que minimice } (L_s(w) + \lambda||w||^2)$$

Este tipo de función de regularización se conoce como egularización de Tikhonov.
Regresión Ridge
Aplicando la regla RLM con regularización de Tikhonov a la regresión lineal con la pérdida al cuadrado, obtenemos la siguiente regla de aprendizaje:

$$arming_{w\in\mathbb{R}^d}\left(\lambda||w||_2^2 + \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}(\langle w, x_i\rangle - y_i)^2\right)$$

Realizando una regresión lineal con la ecuación anterior se le denomina regresión ridge.

Para resolver la ecuación, comparamos el gradiente de la ecuación a cero y obtenemos un conjunto de ecuaciones lineales

$$(2\lambda mI + A)w = b$$

donde I es la matriz identidad y A, b son definidos como se definieron con anterioridad en la página 11 y 12.

Como Aes definida como una matriz semipositiva y la atriz $2\lambda mI+A$ tiene sus eigenvalores acotados por debajo por $2\lambda m$, entonces la matriz es invertible y la solución está dada por :

$$w = (2\lambda I + A)^{-1}b$$

1.6. Árboles de decisión

Un árbol de decisión es un diagrama que representa los diversos resultados a los que es posible llegar a través de una serie de decisiones lógicas. En estadística, un árbol de decisión puede funcionar como un modelo de regresión o de clasificación cuyo objetivo es clasificar las observaciones en un subconjunto de clases conocidas.

Los árboles de decisión son muy eficientes y pueden modelar escenarios extremadamente complejos, adicionalmente, con ellos se obtienen resultados rápidamente. Los árboles de decisión son considerados como el algoritmo más interpretable en inteligencia artificial; los resultados que arroja un árbol de decisión están al alcance de personas sin conocimientos en el tema; esto último le da a loa árboles de decisión una gran ventaja sobre otros modelos.

Un árbol de decisión, consta de nodos conectados por aristas dirigidas. Podemos pensar en el árbol como la representación en forma de diagrama de flujo de una función $h : \mathcal{X} \to \mathcal{Y}$ tal que a cada vector de características $x \in \mathcal{X}$ le asigna una etiqueta $h(x) \in \mathcal{Y}$ (misma que puede ser continua o discreta). La idea es que el árbol de decisión esboce un manual paso a paso, una receta, de cómo calcular el valor de $h(x)$ dado el vector de características $x \in \mathcal{X}$. El cálculo de dicho valor comienza en el nodo raíz del árbol y finaliza en uno de los nodos terminales del mismo.

Existen dos tipos de nodos en un árbol de decisión: los nodos internos o de decisión y los nodos hoja (del inglés *leaf nodes*). Por un lado, los nodos de decisión, como su nombre lo indica, representan *tests* particulares sobre el vector de características x, mismos que serán considerados para tomar una decisión y que se encuentran en función de una determinada condición establecida en el nodo del que provienen. Por otro lado, los nodos terminales (a los que denotaremos por $\widehat{y}$) representan el resultado final de toda una serie de decisiones; los nodos terminales se caracterizan por no tener aristas salientes.

Debido a nuestra intención de "predecir" el valor de $h(x)$ para una $x \in \mathcal{X}$ dada con el árbol de decisión, cada nodo terminal del mimo $\widehat{y}$ representará a lo que denominaremos una región de decisión $R_{\widehat{y}}$, que no es más que un subconjunto del espacio de características ($R_{\widehat{y}} \subseteq \mathcal{X}$). Así, la hipótesis h asociada con un árbol de decisión será constante sobre las regiones $R_{\widehat{y}}$, esto es que para cierto $\widehat{y}$ se satisface que $h(x) = \hat{y}$ para toda $x \in R_{\widehat{y}}$.

Ejemplo 1.

En la Figura 5 se muestra un árbol de decisión; en este caso consideraremos $X = \mathbb{R}^2$. El árbol de decisión de la Figura 5 induce una partición del plano en tres regiones de decisión (ver Figura 6). Es claro que cada región de decisión $R_{\widehat{y}}$ está asociada a un nodo terminal $\widehat{y}$ del árbol de decisión.

En este ejemplo, podemos predecir la hipótesis $h(x)$ para toda $x \in \mathcal{X}$ en función de los vectores u, v y un radio positivo ϵ. Decimos en este caso que el espacio hipótesis queda parametrizada por los vectores u, v y el escalar ϵ.

Definiremos la profundidad de un árbol de decisión como el máximo número de saltos que toma alcanzar un nodo terminal partiendo desde el nodo raíz y siguiendo las flechas de las aristas del árbol. Bajo la anterior definición, el árbol del Ejemplo 1 tiene profundidad 2. Debido a las limitaciones de los recursos computacionales, sólo es posible utilizar árboles

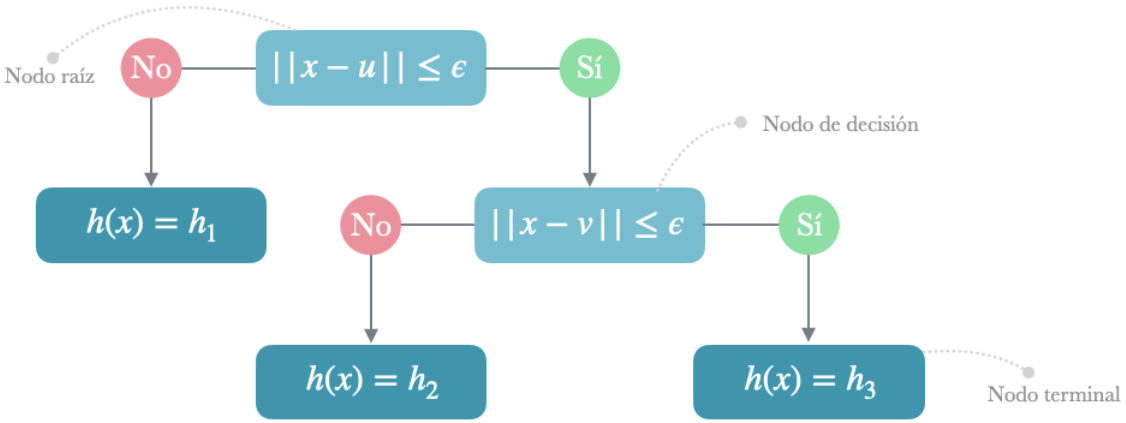

Figura 5: Ejemplo de un árbol de decisión.

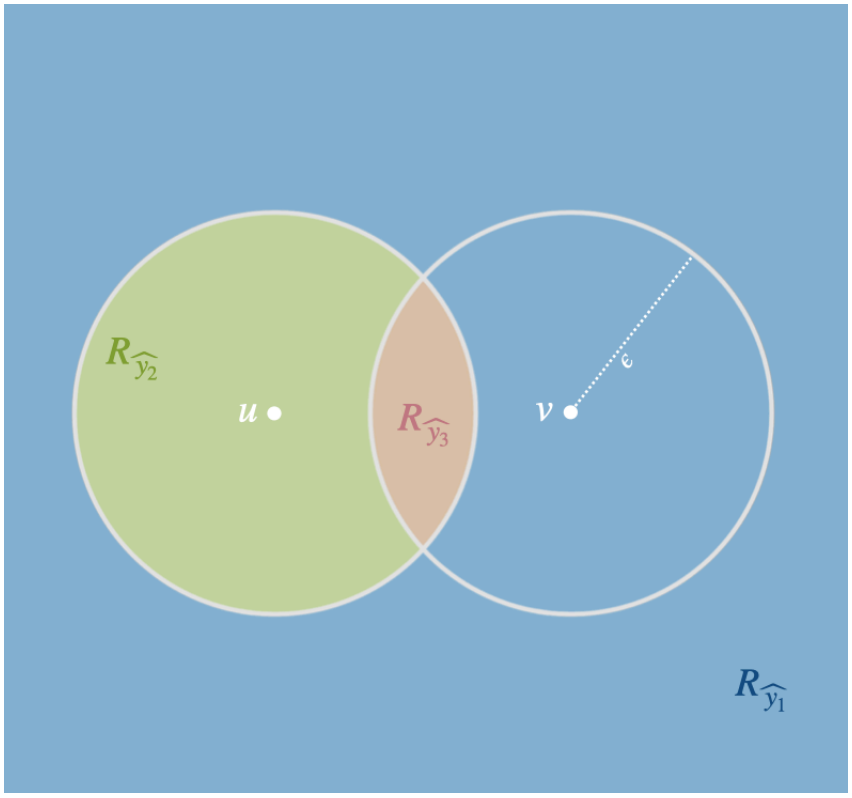

Figura 6: Regiones de decisión inducidas por el árbol de decisión del ejemplo.

de decisión con profundidad limitada.

Como el lector podrá notar, un árbol de decisión representa un mapeo $h : \mathcal{X} \to \mathcal{Y}$ que es constante sobre regiones disjuntas (las regiones de decisión) del espacio de características $\mathcal{X}$. Así, estas regiones particionan el espacio de características en subconjuntos de características que son mapeadas a la misma etiqueta de predicción; como ya se ha mencionado, cada nodo terminal del árbol de decisión corresponde a una región de decisión particular. Utilizando grandes árboles de decisión con múltiples nodos de decisión podemos representar una hipótesis h con regiones de decisión complicadas; mismas que pueden ser elegidas de tal modo que se alineen de manera perfecta con cualquier conjunto de datos etiquetado (ver Ejemplo 2).

Ejemplo 2.

Utilizando un árbol de decisión suficientementemente grande (profundo), podemos construir h de forma que se ajuste a cualquier conjunto de datos $\{(x^{(i)}, y^{(i)}\}_{i=1}^{m}$ tal que $h(x^{(i)}) = y^{(i)}$ para $i = 1, ..., m$. Así, si por ejemplo $\mathcal{X} = \mathbb{R}^2$ y $m = 4$, el árbol de decisión de la Figura 7 funciona para nuestro propósito (ver Figura 8).

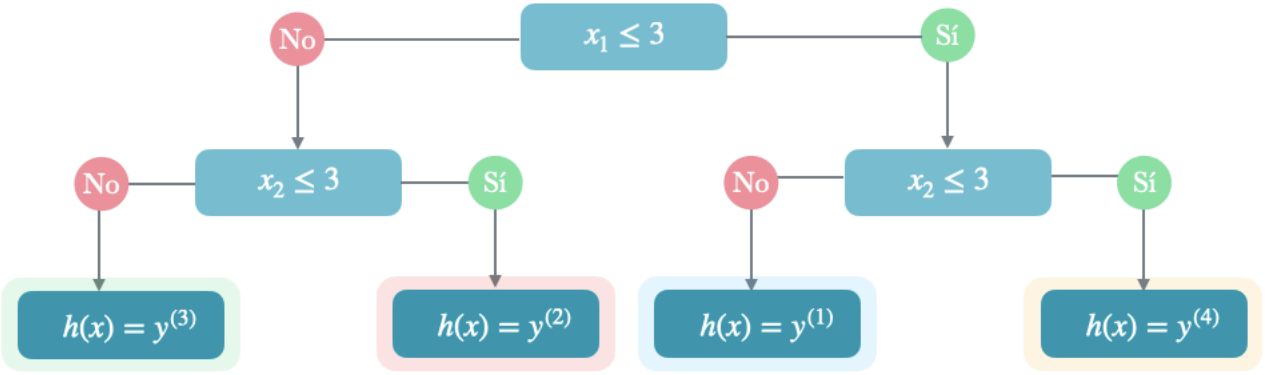

Figura 7: Árbol de decisión.

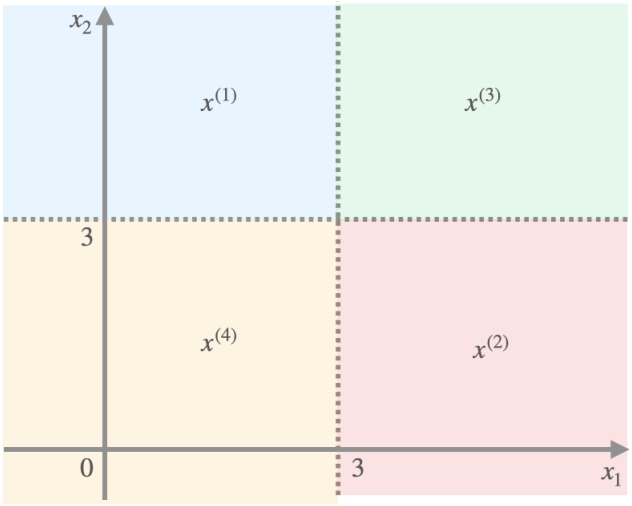

Figura 8: Partición del espacio de características $\mathcal{X} = \mathbb{R}^2$ en las regiones de decisión inducidas por el árbol de decisión.

Ejemplo 3.

Consideremos ahora el ejemplo de cómo se identificó en el Centro Médico de San Diego a pacientes de alto riesgo, donde con pacientes de alto riesgo nos referimos a aquellos que no sobrevivirán al menos 30 días después de haber presentado un paro cardiaco, tomando como referencia los datos de las primeras 24 horas después de que el paciente presentó un paro cardiaco. Ver figura 9.

Es claro que en este ejemplo, la variable respuesta es binaria: alto riesgo (0) y bajo riesgo (1).

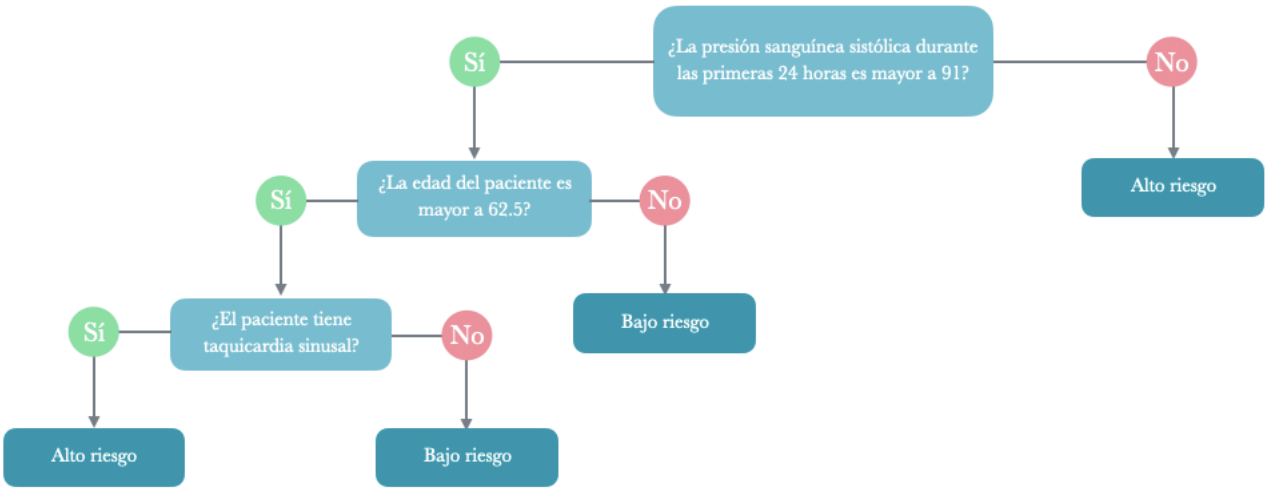

Figura 9: Árbol de decisión para la clasificación de pacientes de alto riesgo.

Supongamos que en en lugar de preguntarnos si el paciente es o no de alto riesgo nos interesara, por ejemplo, el número de días esperado que el paciente podrá sobrevivir. Bajo este escenario, el árbol de decisión asociado al nuevo problema cambiará: los nodos terminales indicarán ahora el promedio esperado de los días que el paciente sobrevivirá. Esta situación describe un árbol de regresión, en lugar de un árbol de clasificación (como los que habíamos estado trabajando hasta ahora). En general, en un árbol de clasificación se busca clasificar a las observaciones en categorías, mientras que en un árbol de regresión se intenta predecir valores numéricos.

Formalicemos un poco lo que hemos trabajado hasta ahora. Sea Y la variable dependiente (categórica o continua) y $x \in \mathcal{X} = \mathbb{R}^d$. Nos interesa obtener

$$h(x) = E[Y|X = x]$$

Los árboles de decisión (de regresión o clasificación) construyen la función h a través de una función escalón, cuyos saltos a lo largo de los ejes coordinados deben determinarse a partir de los datos. En la figura 10 se muestra un ejemplo de esto.

Para evaluar la calidad de un árbol de decisión en particular, podemos usar diversas funciones de pérdida (*loss functions*). Por ejemplo, Error Cuadrático Medio (ECM) para etiquetas numéricas (regresión) o la función de impureza de cada región de decisión para etiquetas categóricas (clasificación).

Antes de continuar, discutamos un par de cuestiones. Existen muchas maneras de construir un árbol de decisión: podemos elegir de múltiples formas qué característica vamos a poner como condición en cada nodo del árbol. El objetivo es realizar esta selección de manera que nuestro árbol clasifique los datos con buena precisión. Para entender mejor el problema asociado a la construcción de un árbol, vamos a poner un ejemplo.

Ejemplo 5.

Consideremos el conjunto de datos descrito en el Cuadro 2. Este dataset nos dice que bebidas nos enferman. Vamos a construir, visualmente un árbol de decisión.

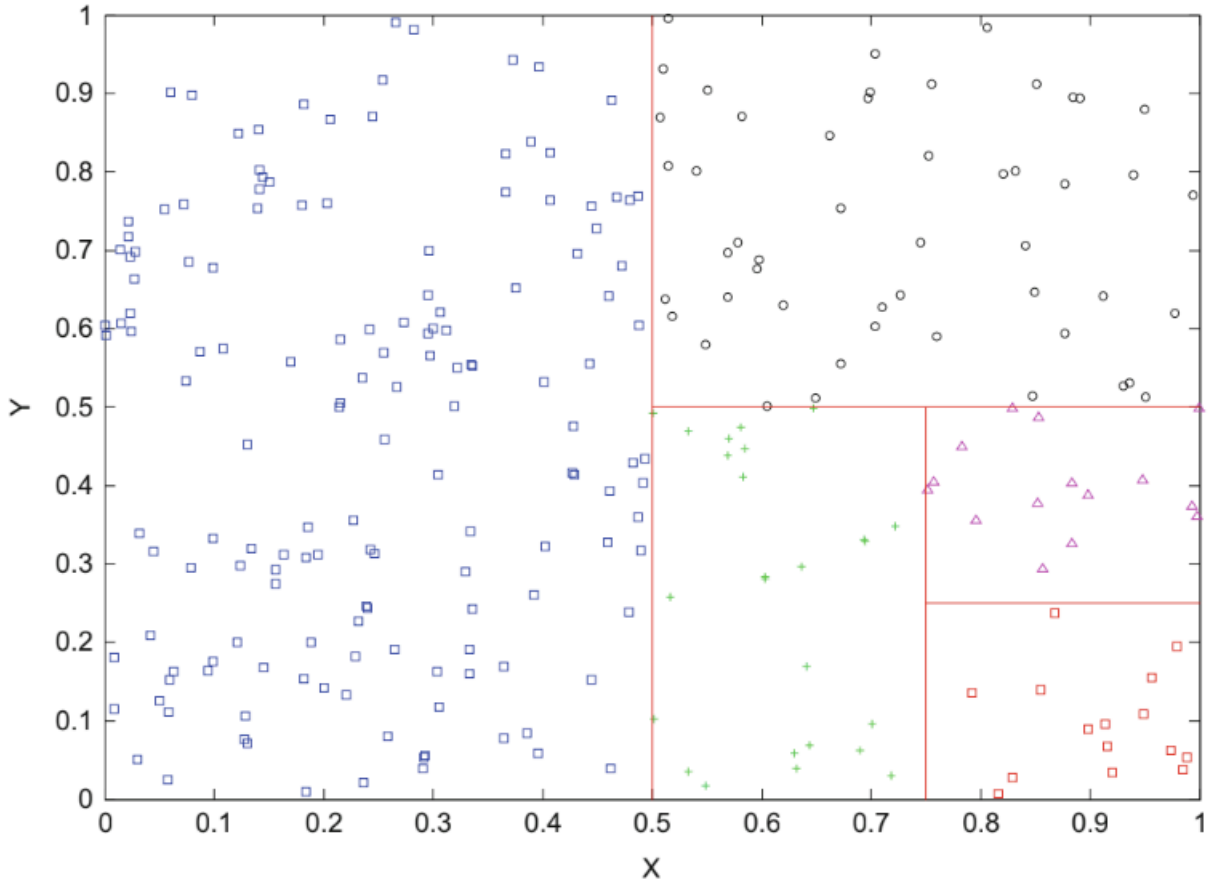

Figura 10: Ejemplo de las regiones de decisión, cada color corresponde a un cluster.

Forma	Color	Contenido	¿Causó enfermedad?
Cilindro	Naranja	25cl	No
Cilindro	Negro	25cl	No
Cono	Blanco	10cl	No
Trapezoide	Verde	15cl	No
Cono	Amarillo	15cl	No
Trapezoide	Naranja	15cl	Sí
Cono	Naranja	15cl	Sí
Cono	Naranja	15cl	Sí

Cuadro 2: Conjunto de datos

Figura 11: Descripción visual del conjunto de datos

La primer observación pertinente es que las bebidas cilíndricas no causan enfermedad (por lo que la forma es relevante para la clasificación, ver Figura 12). Ahora, una segunda observación importante es que de entre las bebidas que no son de forma cilíndrica, las de color naranja resultan seguras (por lo que el color es también relevante, ver Figura 13).

Figura 12: La forma es una característica relevante

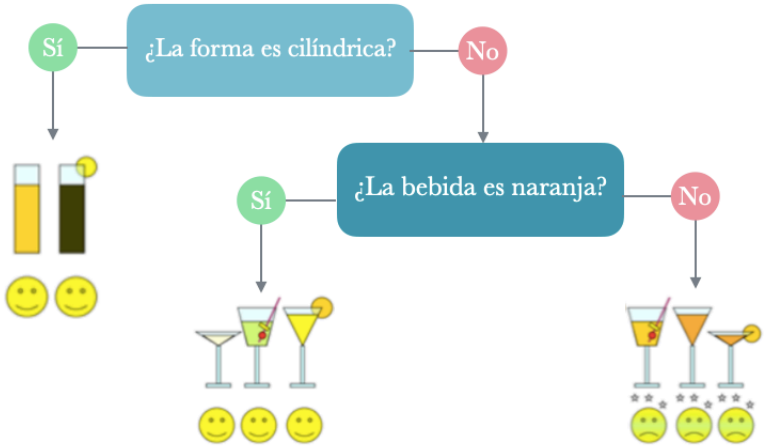

Figura 13: El color es una característica relevante

Así, hemos generado un árbol de decisión a partir del dataset, que nos permite distinguir las bebidas inseguras de las seguras, ver Figura 14. Ahora bien, en este caso generamos el árbol de decisión *a ojo*, ¿cómo podemos hacerlo matemáticamente? Necesitamos decidir qué criterio de decisión poner en cada nodo del árbol, para hacerlo se suele hacer uso del índice de Gini (que discutiremos a continuación).
Implementación de un árbol de clasificación: Medidas de impureza.
Como ya hemos discutido, al construir un árbol de clasificación, uno debe preguntarse qué decisiones deben ponerse en el árbol, es decir, bajo qué criterio deberíamos dividir a los datos de modo que la clasificación hecha por el árbol de decisión sea buena. Es justo este problema el que discutiremos a lo largo de esta sección.

Supóngase que se tienen n observaciones en la muestra de aprendizaje y que n_j es el número total de observaciones pertenecientes a la clase j, con $j = 1, \ldots, J$. Así, las probabilidades de clase son

$$\pi(j) = \frac{n_j}{n} \quad j = 1, \ldots, J \tag{10}$$

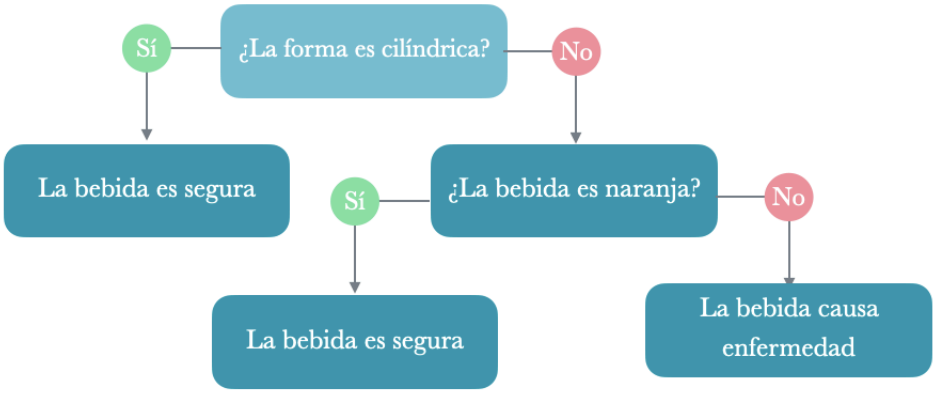

Figura 14: Árbol de decisión resultante

Es decir, $\pi(j)$ es la proporción de observaciones pertenecientes a una clase en particular. Sea $n(t)$ el número de observaciones en el nodo t y $n_j(t)$ el número de observaciones en el nodo t que pertenecen a la clase j-ésima. Así, la probabilidad del evento en que una observación de la clase j-ésima cae en el nodo t es:

$$p(t,j) = \pi(j)\frac{n_j(t)}{n_j} \tag{11}$$

Así, la proporción de observaciones en el nodo t es

$$p(t) = \sum_{j=1}^{J} p(t,j) \tag{12}$$

Observemos que si sustituimos (10) en (11)

$$p(t,j) = \pi(j)\frac{n_j(t)}{n_j} = \frac{n_j}{n}\cdot\frac{n_j(t)}{n_j} = \frac{n_j(t)}{n} \tag{13}$$

Y si sustituimos (13) en (12)

$$p(t) = \sum_{j=1}^{J} p(t,j) = \sum_{j=1}^{J} \frac{n_j(t)}{n} = \frac{1}{n}\sum_{j=1}^{J} n_j(t) = \frac{n(t)}{n} \tag{14}$$

Luego, por (13) y (14), la probabilidad condicional de que una observación pertenezca a la clase j dado que está en el nodo t es

$$p(j|t) = \frac{p(j,t)}{p(t)} = \frac{n_j(t)}{n(t)} \tag{15}$$

Definiremos ahora el grado de homogeneidad de clase en un nodo dado. Esta característica —una medida de impureza $i(t)$— representará un indicador para la homogeneidad de una clase en un nodo del árbol espcificado y por tanto, nos ayudará a encontrar los puntos de división óptimos (buscaremos maximizar la homogeneidad de las clases en el nodo).

Defínase una función de impureza $\iota(t)$ como una función definida sobre $(p_1,\ldots,p_J) \in [0,1]^J$ con $\sum_{j=1}^{J} p_j = 1$, tal que:

- ι tiene un único máximo en $\left(\frac{1}{J},\frac{1}{J},\ldots,\frac{1}{J}\right)$;

- ι tiene un único mínimo en $(1, 0, \ldots, 0),\ (0, 1, \ldots, 0)\ , \ldots,\ (0, 0, \ldots, 1)$;
- ι es una función simétrica de $p_1, \ldots, p_J$.

Toda función que satisfaga las condiciones anteriores se llama una *función de impureza*. Dado un nodo t del árbol, se define una medida de impureza $i(t)$ para dicho nodo como sigue

$$i(t) = \iota\{p(1|t), p(2|t), \ldots, p(J|t)\}$$

Denotemos a una división arbitraria de los datos por s, así, para un nodo dado t (al que nos referiremos por nodo padre) surgen dos nodos hijos: t_L y t_r, representando las observaciones que cumplen y no cumplen el criterio de división s, respectivamente. Ver figura 15. Una fracción p_L de los datos del nodo t cae en el hijo izquierdo del nodo t (en t_L) y la fracción restante $p_R = 1 - p_L$ en el hijo derecho t_R.

Figura 15: Jerarquía entre el nodo padre y los nodos hijos en un árbol de decisión.

Una medida de la calidad de la división generada por s está dada por $\Delta i(s, t)$, que se define como

$$\Delta i(s, t) = i(t) - p_L i(t_L) - p_R i(t_R) \tag{16}$$

Entre mayor sea el valor de $\Delta i(s, t)$, mejor será el criterio de división s, puesto que la impureza de los datos se redujo. Para hallar el criterio óptimo de división s^*, el procedimiento natural sería entonces maximizar $i(s, t)$ como función de s. Obsérvese que en (16), para diferentes divisiones s, el valor de $i(t)$ permanece constante, por lo que

$$\begin{aligned} s^* &= \underset{s}{\text{argmax}}\ \Delta i(s, t) \\ &= \underset{s}{\text{argmax}}\ \{-p_L i(t_L) - p_R i(t_R)\} \\ &= \underset{s}{\text{argmin}}\ \{p_L i(t_L) + p_R i(t_R)\} \end{aligned}$$

Es claro que t_L y t_R son funciones implícitas de s.

Este proceso de división se repite hasta que uno llega a un *bucket size* mínimo. Una vez alcanzado este punto, las clases se asignan a los nodos terminales siguiendo la regla que se presenta a continuación.

$$\text{Si } p(j|t) = \underset{k}{\text{máx}}\ p(k|t),\ \text{entonces } j^*(t) = j$$

Si el máximo no es único, entonces a $j^*(t)$ se le asigna aleatoriamente alguna de las clases para las cuales $p(k|t)$ se maximiza.

La cuestión crucial es determinar una medida de impureza $i(t)$. Una definición natural de *impureza* es a través de la varianza: se asigna 1 a todas las observaciones del nodo t que pertenezcan a la clase j y 0 a las otras. Siendo así, un estimado para la varianza de la muestra de las observaciones del nodo t es

$$p(j|t)\{1 - p(j|t)\}$$

Sumando sobre las J clases, obtenemos el índice de Gini

$$i(t) = \sum_{j=1}^{J} p(j|t)\{1 - p(j|t)\} = \sum_{j=1}^{J} \{p(j|t) - p(j|t)^2\} = 1 - \sum_{j=1}^{J} p(j|t)^2$$

El índice de Gini es una función de impureza $\iota(p_1, p_2, \ldots, p_J)$ si tomamos $p_j = p(j|t)$. No es difícil verificar que el índice de Gini es una función cóncava. Como $p_L + p_R = 1$, obtenemos

$$\begin{aligned} & p_L i(t_L) + p_R i(t_R) \\ & = p_L \iota\{p(1|t_L), p(2|t_L), \ldots, p(J|t_L)\} + p_R \iota\{p(1|t_R), p(2|t_R), \ldots, p(J|t_R)\} \\ & \leq \ \iota\{p_L p(1|t_L) + p_R p(1|t_R), \ldots, p_L p(J|t_L) + p_R p(J|t_R)\} \end{aligned}$$

por definición de función cóncava; en donde la desigualdad se convierte en igualdad cuando $p(j|t_L) = p(j|t_R)$ para cada $j \in \{1, \ldots, J\}$.

Observemos que

$$\begin{aligned} \frac{p(j, t_L)}{p(t)} &= \frac{p(t_L)}{p(t_L)} \cdot \frac{p(j, t_L)}{p(t)} \\ &= \frac{p(t_L)}{p(t)} \cdot \frac{p(j, t_L)}{p(t_L)} \\ &= p_L p(j|t_L) \end{aligned}$$

donde la última igualdad se da por definición de p_L y por (15). Análogamente,

$$\frac{p(j, t_R)}{p(t)} = p_R p(j|t_R)$$

Luego, como los datos en t son la unión de los datos de t_R y t_L (y ningún dato está en ambos nodos a la vez),

$$\begin{aligned} p(j|t) &= \frac{p(j, t)}{p(t)} \\ &= \frac{p(j, t_L) + p(j, t_R)}{p(t)} \\ &= p_L p(j|t_L) + p_R p(j|t_R) \end{aligned}$$

Así, como

$$p_L i(t_L) + p_R i(t_R) \ \leq \ \iota\{p_L p(1|t_L) + p_R p(1|t_R), \ldots, p_L p(J|t_L) + p_R p(J|t_R)\}$$

por la anterior igualdad concluimos que

$$p_L i(t_L) + p_R i(t_R) \ \leq \ \iota\{p(1|t), \ldots, p(J|t)\} = i(t)$$

Con lo anterior, es fácil ver en (16) que $\Delta i(s,t)$ será positivo salvo cuando $p(j|t_L) = p(j|t_R)$ para cada $j \in \{1,\ldots,J\}$, es decir, cuando una división no reduzca la heterogeneidad de la clasificación.

Las medidas de impureza pueden definirse de múltiples maneras, explicaremos a continuación otra de ellas: la llamada regla *twoing*. En lugar de maximizar el cambio en la impureza en un nodo particular, la regla *twoing* intenta balancear el árbol como si la muestra de aprendizaje tuviera únicamente dos clases. La ventaja de dicho algoritmo es que esta regla de decisión es capaz de distinguir observaciones tomando en cuenta factores generales en los niveles altos del árbol, mientras que para los niveles bajos del mismo toma en cuenta características específicas de los datos para hacer la separación.

Si $S = \{1,2,\ldots,J\}$ es el conjunto de las clases en que pueden ser categorizados los datos de aprendizaje, este se divide en dos subconjuntos de S

$$S_1 = \{j_1,\ldots,j_n\} \text{ y } S_2 = S \setminus S_1$$

Todas las observaciones en S_1 son asignadas a la clase *dummy* 1, el resto es asignado a la clase *dummy* 2. El siguiente paso es calcular $\Delta i(s,t)$ como si hubiese sólo dos clases (las clases *dummy*). Dado que el valor de $\Delta i(s,t)$ depende de la elección de S_1, debemos maximizar $\Delta i(s,t,S_1)$. Posteriormente hay que aplicar un procedimiento de dos pasos (de aquí el nombre *twoing*): primero hallar $s^*(S_1)$ maximizando $\Delta i(s,t,S_1)$ y posteriormente hallar S_1^* maximizando $\Delta i(s^*(S_1),s,S_1)$. En otras palabras, la idea de la regla *twoing* es encontrar una combinación de superclases en cada nodo que maximice el incremento en la medida de impureza para dos clases.

Este método tiene una gran ventaja: encuentra los llamados nodos estratégicos, es decir, nodos que filtran las observaciones de modo que estas queden diferenciadas de la mejor manera posible. Sin embargo, aunque aplicar la regla *twoing* puede parecer deseable (especialmente cuando el conjuntos de datos tiene un número grande de clases), surge un nuevo reto: lidiar con la limitación computacional. Por ejemplo, supóngase que la muestra de aprendizaje tiene J clases, entonces el conjunto S se puede dividir en S_1 y S_2 de 2^{J-1} maneras. Así, para un conjunto de datos con 11 clases, tendríamos más de 1000 combinaciones posibles. Afortunadamente, el siguiente resultado ayuda a reducir drásticamente el número de cálculos realizados.

Se puede probar que en una tarea de clasificación con dos clases y medida de impureza $p(1|t)p(2|t)$, por cada división s una superclase $S_1(s)$ es determinada por

$$S_1(s) = \{j \ : \ p(j|t_L) \geq p(j|t_R)\}$$

donde,

$$\max_{S_1} \Delta i(s,t,S_1) = \frac{p_L p_R}{4}\left\{\sum_{j=1}^{J} |p(j|t_L) - p(j|t_R)|\right\}^2$$

Por lo tanto, al igual que el índice de Gini, la regla *twoing* puede ser aplicada en la práctica. Es importante mencionar que este último criterio siempre es un poco más lento.

Veamos ahora un ejemplo de cómo implementar un árbol de decisión a partir de un conjunto de datos de entrenamiento haciendo uso del índice de Gini.

Ejemplo 6.

Consideremos el conjunto de datos descrito en el Cuadro 3. Cada observación de el dataset corresponde a una persona, este dataset indica si a dicho individuo le gustan las palomitas, el refresco, así como la edad de la persona en cuestión, para posteriormente indicarnos si es que le agrada o no la saga Star Wars. Nuestro objetivo será entonces crear un árbol de decisión a partir de el conjunto de datos dado que intente predecir si a una persona le gusta o no Star Wars.

Gusto por las palomitas	Gusto por el refresco	Edad	Gusto por Star Wars
Sí	Sí	7	No
Sí	No	12	No
No	Sí	18	Sí
No	Sí	35	Sí
Sí	Sí	38	Sí
Sí	No	50	No
No	No	83	No

Cuadro 3: Conjunto de datos

Lo primero que haremos será decidir si en el nodo raíz del árbol deberíamos preguntar por *Edad, Gusto por las palomitas* o *Gusto por el refresco*. Para tomar esta decisión, averiguaremos qué tan bien predice cada covariable a la variable respuesta (*Gusto por Star Wars)*, utilizando como medida de impureza al índice de Gini. Notemos que nuestra variable a predecir es binaria — a la persona le gusta ($j = 1$) o no le gusta Star Wars ($j = 2$) —, por lo que sólo tenemos dos clases, es decir, $J = 2$.

Recordemos que el criterio de división óptimo s^* en las medidas de impureza era aquel que maximizaba $\Delta i(s, t)$ (ver 16), o lo que es lo mismo, el que minimizaba

$$p_L i(t_L) + p_R i(t_R)$$

Estudiaremos con esta métrica qué tan bueno es cada uno de los criterios.

- Criterio de división por ***Gusto por las palomitas.***

  Tomando en cuenta a la covariable *Gusto por el refresco* el criterio de división en el nodo t sería *¿A la persona le gustan las palomitas?*. El nodo hijo izquierdo, t_L, pondremos a aquellas personas a las que sí les gustan las palomitas y en el derecho, t_R, a las que no. Así, según nuestro dataset,

$$p_L = \frac{4}{7} \quad p_R = \frac{3}{7}$$

Y los índice de Gini para t_L y t_R son, respectivamente

$$\begin{aligned} i(t_L) &= 1 - p(1|t_L)^2 - p(2|t_L)^2 \\ &= 1 - \left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2 \\ &= \frac{6}{16} \end{aligned}$$

$$\begin{aligned} i(t_R) &= 1 - p(1|t_R)^2 - p(2|t_R)^2 \\ &= 1 - \left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2 \\ &= \frac{4}{9} \end{aligned}$$

Notemos que $p(1|t_L) = \frac{1}{4}$, pues de entre las 4 personas con gusto a las palomitas, sólo a una le gusta Star Wars. Análogamente, $p(1|t_L) = \frac{2}{3}$ ya que de entre las 3 personas a las que no le gustan las palomitas, a dos les gusta la saga.

De lo anterior se tiene que,

$$p_L i(t_L) + p_R i(t_R) = \frac{4}{7} \cdot \frac{6}{16} + \frac{3}{7} \cdot \frac{4}{9} \approx 0{,}405$$

Analicemos ahora los otros criterios de división.

- Criterio de división por ***Gusto por el refresco.***

  En esta ocasión, el criterio de división en el nodo t sería *¿A la persona le gusta el refresco?* En el nodo hijo izquierdo t_L posicionaremos a aquellas personas a las que les agrada el refresco y en el derecho t_R a las que no. Así, de acuerdo al dataset dado,

$$p(t_L) = \frac{4}{7}$$

$$p(t_R) = \frac{3}{7}$$

$$i(t_L) = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = \frac{6}{16}$$

$$i(t_R) = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

Así,

$$p_L i(t_L) + p_R i(t_R) = \frac{4}{7} \cdot \frac{6}{16} + \frac{3}{7} \cdot 0 \approx 0{,}214$$

- Criterio de división por ***Edad.***

  A diferencia de las dos covariables estudiadas anteriormente, la covariable edad es de tipo continuo; consecuentemente, el criterio de división generado por la *Edad* no es tan claro como en los casos previos. A continuación, explicaremos la manera de proceder. Lo primero que haremos será ordenar las filas del dataset de acuerdo al criterio de edad, ordenaremos de menor a mayor. Posteriormente, calcularemos el promedio de valores adyacentes como se muestra del lado izquierdo de la Figura 16.

  Cada uno de los nuevos valores calculados nos servirá para la creación de un criterio de división. Llamemos x a uno de los promedios adyacentes, el criterio de división asociado a este criterio será

  $$\textit{¿La edad de la persona es menor que } x\textit{?}$$

  Posicionaremos a las personas con edad menor a x en el nodo hijo izquierdo, t_L, y a las restantes en el derecho, t_R. Posteriormente calcularemos la medida de impureza $p_L i(t_L) + p_R i(t_R)$ para cada x.

  Así, por ejemplo, para $x = 9{,}5$,

  $$p(t_L) = \frac{1}{7}$$

  $$p(t_R) = \frac{6}{7}$$

  $$i(t_L) = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 = 0$$

  $$i(t_R) = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = \frac{1}{2}$$

  Así,

  $$p_L i(t_L) + p_R i(t_R) = \frac{1}{7} \cdot 0 + \frac{6}{7} \cdot \frac{1}{2} \approx 0{,}429$$

  En el lado derecho de la Figura 16 se muestra el resultado de este cálculo para cada valor x.

  Como se puede observar en la Figura 16, los criterios de división que minimizan $p_L i(t_L) + p_R i(t_R)$ son

  $$\textit{¿La edad de la persona es menor que 15?}$$
  $$\textit{¿La edad de la persona es menor que 44?}$$

  con un valor de 0,343. Elegiremos arbitrariamente el primero.

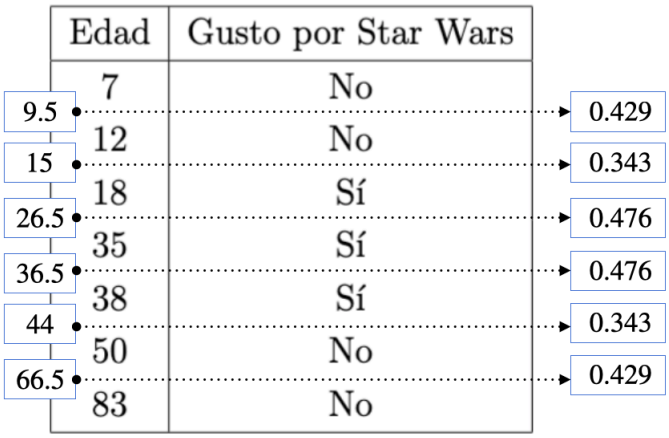

Figura 16: Cálculo del índice de Gini para valores adyacentes de la variable continua.

Criterio de división	$p_L i(t_L) + p_R i(t_R)$
¿A la persona le gustan las palomitas?	0,405
¿A la persona le gusta el refresco?	0,214
¿La edad de la persona es menor que 15?	0,343

Cuadro 4: Criterios de decisión y la medida de impureza asociada a ellos

En el Cuadro 4 se mencionan los criterios de división estudiados así como el valor de $p_L i(t_L) + p_R i(t_R)$ asociado a ellos. Es claro que el que minimiza $p_L i(t_L) + p_R i(t_R)$ es *¿A la persona le gusta el refresco?*
Siendo así, según lo estudiado anteriormente, concluimos que el árbol de decisión debe tener en su nodo raíz al criterio de división *¿A la persona le gusta el refresco?* Como ya vimos, el nodo hijo derecho tiene una medida de impureza 0, por lo que ya no es necesario seguir agregando ramas al árbol de ese lado. Por otro lado, en el nodo izquierdo la impureza es de $\frac{6}{16}$; así, podemos repetir el proceso anterior para seguir agregando nodos de decisión con el objetivo de reducir la impureza, esta vez trabajando sobre el subconjunto de datos constituido por aquellas personas a las que sí les gusta el refresco (ver Cuadro 5).

Gusto por las palomitas	Gusto por el refresco	Edad	Gusto por Star Wars
Sí	Sí	7	No
No	Sí	18	Sí
No	Sí	35	Sí
Sí	Sí	38	Sí

Cuadro 5: Subconjunto del dataset original constituido por aquellas personas a las que sí les gusta el refresco.

Los criterios de decisión a evaluar, así como el valor de $p_L i(t_L) + p_R i(t_R)$ asociado a ellos, se contrastan en el cuadro 6. Es claro que el mejor criterio es *¿La edad de la persona es menor que 12.5?*.
Dado que el valor de $p_L i(t_L) + p_R i(t_R)$ es cero para el criterio escogido es cero, no tiene sentido seguir agregando ramas al árbol. Por lo que lo único que resta es agregar nodos terminales. A cada nodo terminal, naturalmente, le asignaremos el valor de la categoría que presente la mayor frecuencia dentro del dataset de entrenamiento. El árbol de decisión resultante se muestra en la Figura 17.

Criterio de división	$p_L i(t_L) + p_R i(t_R)$
¿A la persona le gustan las palomitas?	0,25
¿La edad de la persona es menor que 12.5?	0

Cuadro 6: Criterios de decisión y la medida de impureza asociada a ellos

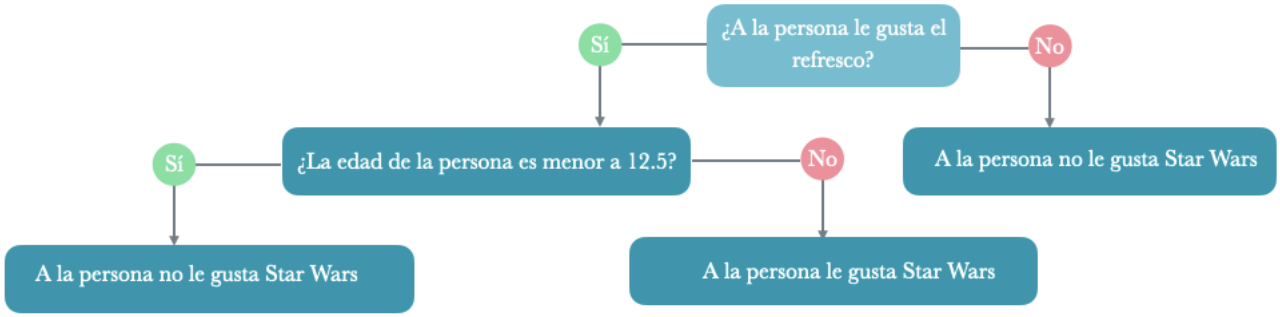

Figura 17: Árbol de decisión resultante

Es importante mencionar que del conjunto de datos dado, sólo una persona cae en el primer nodo terminal (leyendo el árbol de izquierda a derecha), por lo que es difícil asegurar que nuestro árbol hará buenas predicciones cuando se le presente un conjunto de datos distinto al dataset de entrenamiento (overfitting). Discutiremos este problema en la siguiente sección.

Tamaño óptimo de un árbol de decisión

Hasta ahora hemos discutido el problema de encontrar la mejor división s^* en un nodo particular. La siguiente pregunta relevante es cómo determinar el tamaño óptimo del árbol de decisión, es decir, cuándo dejar de hacer divisiones en los nodos. Si cada nodo terminal tiene sólo clases homogéneas del conjunto de datos de entrenamiento, entonces cada elemento de este conjunto puede ser perfectamente clasificado por este *árbol máximo*, pero, ¿es este enfoque fructífero? El árbol máximo es un caso de *overfitting*.

Necesitamos un criterio que nos permita decidir cuándo dejar de ramificar el árbol de decisión. Como la construcción del árbol depende de $\Delta i(s,t)$, un posible criterio es dejar de ramificar si

$$\Delta i(s,t) < \overline{\beta}$$

en donde $\overline{\beta}$ es un valor que funciona como umbral.

Desafortunadamente, el valor de $\overline{\beta}$ se escoge de una manera subjetiva al no haber un método para su cálculo. Casos empíricos muestran que frecuentemente el incremento en la pureza no es monótono, por esta razón, incluso para una $\overline{\beta}$ podemos caer en un caso de *underfitting*. Si consideramos valores de $\overline{\beta}$ aún más pequeños, probablemente este problema se solucione, sin embargo el costo puede ser un caso de *overfitting*.

Otro método para determinar el tamaño adecuado de un árbol de decisión consiste en pedir un mínimo número de observaciones $\overline{N}$ (*bucket size*) en cada nodo terminal. Una desventaja de este criterio es que si en un nodo terminal t el número de observaciones es

mayor que $\overline{N}$, esto es que

$$N(t) > \overline{N}$$

entonces este nodo se vuelve a ramificar, pues bajo este criterio los datos todavía no están lo suficientemente bien separados.

1.6.1. Árboles de regresión.

Hasta ahora nos hemos concentrado en los árboles de clasificación. Si bien los árboles de regresión comparten un esquema lógico bastante similar, existen diferencias entre ambos modelos que vale la pena mencionar. La principal diferencia entre los árboles de clasificación y los de regresión es el tipo de variable dependiente Y. Cuando Y es discreta, el árbol de decisión es un árbol de clasificación, y el árbol es de regresión cuando la variable dependiente Y es continua.

En el índice de Gini y la regla *twoing* (ver Sección 1.6), se asume que el número de clases es finito y por lo tanto, se introducen medidas basadas principalmente en $p(j|t)$ para una clase arbitraria j y un nodo t cualquiera del árbol de decisión. En el caso en que la variable dependiente Y es continua no hay clases, por lo que el enfoque de estos dos métodos no puede ser utilizado a menos que se agrupen de manera efectiva valores continuos creando clases artificialmente.

Similarmente a como hicimos con el caso discreto, podremos describir la homogeneidad absoluta una vez que definamos una medida adecuada de impureza para los árboles de regresión.

Recordando la idea subyacente del índice de Gini, usar la varianza como un indicador de pureza resulta natural. Como para cada nodo la varianza puede ser fácilmente calculada, entonces el criterio de escisión para un nodo arbitrario t puede escribirse como

$$s^* = \underset{s}{\operatorname{argmax}} \left[p_L \operatorname{var}\{t_L(s)\} + p_R \operatorname{var}\{t_R(s)\} \right]$$

en donde t_l y t_R son los nodos hijos resultantes que claramente dependen de la elección de s.

Entonces el árbol de regresión máximo puede ser fácilmente definido como una estructura en la que cada nodo tiene sólo los mismos valores predichos. Es importante señalar que como un conjunto de datos de naturaleza continua tiene probabilidades mucho más altas de tomar valores distintos comparado con un conjuntos de datos provenientes de variables discretas, el tamaño del árbol de regresión máximo generalmente es muy grande.

1.7. K vecinos más cercanos (KNN)

1.7.1. Introducción

El algoritmo K-vecinos más cercanos, abreviado como KNN por sus siglas en inglés (K - Nearest Neighbor) es de tipo supervisado. Se utiliza para problemas de clasificación como de regresión, aunque principalmente se utiliza para el primero.

El algoritmo KNN es uno de los más simples, almacena todos los datos disponibles y clasifica un nuevo punto de datos en función de la similitud. Esto significa que cuando aparecen nuevos datos, se pueden clasificar fácilmente en una categoría dentro del conjuntos de categorías utilizando el algoritmo.Es un algoritmo no paramétrico, lo que significa que no hace ninguna suposición sobre los datos subyacentes.
[colback=gray!4!white,colframe=blue!40!white!90!green!90!cyan]
Observación.

El algoritmo KNN de conoce como perezoso porque no aprende del conjunto de entrenamiento inmediatamente, sino que almacena el conjunto de datos y, en el momento de la clasificación, realiza una acción en el conjunto de datos.

En la fase de entrenamiento solo almacena el conjunto de datos y cuando obtiene nuevos datos, los clasifica en una categoría que es muy similar a los nuevos datos.
Supongamos que deseamos clasificar la imagen de una criatura, queremos saber si es un gato o un perro. Para esto, podemos usar el algoritmo KNN, ya que funciona en una medida de similitud. Nuestro modelo KNN encontrará las características similares del nuevo conjunto de datos a las imágenes de perros y gatos y, en función de las características más similares, las colocará en la categoría de perros o gatos.

1.7.2. Algoritmo

Clasificación

Para implementar el algoritmo KNN en clasificación, se siguen una serie de pasos. Supongamos que queremos clasificar un nuevo punto, digamos x. Sea X el conjunto de datos almacenados, es decir, aquellos que ya están clasificados y m la cardinalidad de X.

Sea $n \in \mathbb{N}^+, n \leq m$ y $\mathcal{C} = \{c_1, ..., c_n\}$ el conjunto de categorías, es decir, tenemos n categorías diferentes, donde para $j \in \{1, .., n\}$, $c_j \subseteq X$ y para toda $i, j \in \{1, ..., n\}, i \neq j$, $c_j \cap c_j =$ y $\bigcup_{i=1}^{n} c_i = X$. Esto es, que los puntos en X se encuentran todos clasificados, además, que las categorías son excluyentes (un punto no puede pertenecer a más de una categoría). De esta manera, queremos averiguar a cual categoría pertenece el nuevo punto x.

Paso 1. Se seleccionar el valor $k \in \mathbb{N}, k \leq m$ del número de vecinos para x

Paso 2. Se calcula la distancia euclidiana de x a todos los datos que se encuentran en X.

Paso 3. Nos quedamos con los K vecinos más cercanos a x según la distancia euclidiana calculada. Supongamos que son el conjunto $\mathcal{K} = \{x_1, ..., x_k\}$

Paso 4. Entre estos k vecinos, contamos el número de puntos de datos en cada categoría. De esta manera, definamos $C : \mathcal{C} \to \mathbb{N}$ tal que para $x_j \in \mathcal{C}$,

$$C(c_j) = t$$

donde $j \in \{1, ..., n\}$ y t es igual al número de datos t que pertenecen a la categoría c_j en el conjunto $\mathcal{K}$.

Paso 5. Se calcula

$$\max_{i \in \{1,...,n\}} C(c_i)$$

Es decir, en el conjunto $\mathcal{K}$ vemos cuál categoría predomina.

Paso 6. Después, se dice que $x \in c_j$ con $j \in \{1, ..., n\}$ si $\text{máx}_{i \in \{1,...,n\}} C(c_i) = C(c_j)$.

Es decir, que clasificamos al nuevo punto x en la categoría c_j si la cantidad de elementos pertenecientes a la categoría j en el conjunto $\mathcal{K}$ es la más grande respecto a la cantidad que hay de elementos de las otras categorías

Regresión

El concepto general de KNN para la regresión es el mismo que para la clasificación: encontramos los k vecinos más cercanos en el conjunto de datos y luego hacemos una predicción basada en las etiquetas de los k vecinos mas cercanos. Sin embargo, en la regresión, la función objetivo es real en lugar de función de valor discreto.

Un enfoque común para calcular el dato continuo es calcular la media sobre los k vecinos más cercanos, o bien, también es común usar la mediana en vez de la media.

¿Cómo seleccionar el valor de k?

El valor de k en el algoritmo KNN define cuántos vecinos se verificarán para determinar la clasificación de un punto de consulta específico. Por ejemplo, si k=1, la instancia se asignará a la misma clase que su vecino más cercano.

No existe una forma particular de determinar el mejor valor para k, por lo que debemos probar algunos valores para encontrar lo mejor de ellos. Definir k puede ser un acto de equilibrio ya que diferentes valores pueden llevar a un ajuste excesivo o insuficiente.

El valor más preferido para k es 5. Un valor muy bajo para k, como $k = 1$ o $k = 2$, puede ser ruidoso y provocar efectos atípicos en el modelo.Los valores grandes de K son buenos, pero pueden encontrar algunas dificultades, es decir, los valores más bajos de k pueden tener una varianza alta, pero un sesgo bajo, y los valores más grandes de k pueden generar un sesgo alto y una varianza más baja.

La elección de k dependerá en gran medida de los datos de entrada, ya que los datos con más valores atípicos o ruido probablemente funcionarán mejor con valores más altos de k. En general, se recomienda tener un número impar para k para evitar empates en la clasificación, y las tácticas de validación cruzada pueden ayudarlo a elegir la k óptima para su conjunto de datos.

En la Figura 18 deseamos, se tienen 3 categorías : círculos, triángulos y cruces. Se desea clasificar un nuevo dato en alguna de estas categorías, se toman los $k = 5$ vecinos más cercanos y se observa que hay 3 pertenecientes a la categoría de triángulos, 1 a círculo y 1 a cruces, por lo tanto, el dato pertenece a la categoría de triángulos.

1.7.3. ¿Cuándo usar el algoritmo de KNN?

Si bien las redes neuronales están ganando popularidad en la visión por computadora y el reconocimiento de patrones campo, un área donde los modelos de k-vecinos más cercanos todavía son comunes y exitosos se utiliza en la intersección entre la visión artificial, la clasificación de patrones y la biometría (por ejemplo, para hacer predicciones basadas en características geométricas).

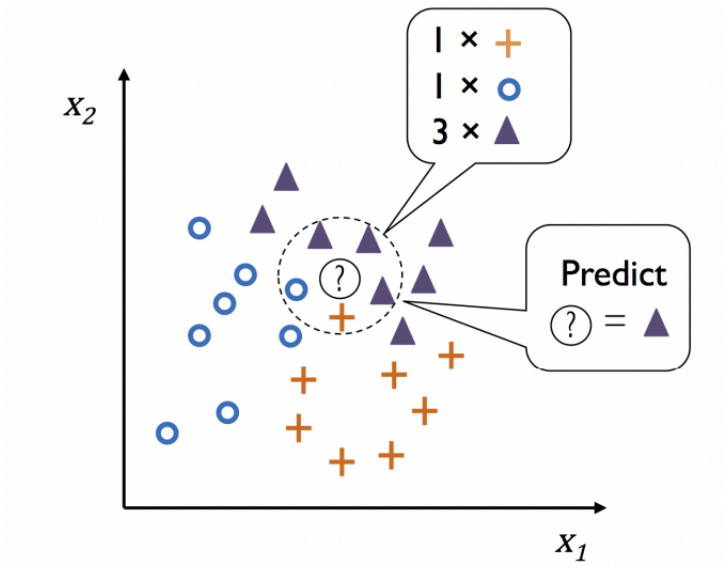

Figura 18: Algoritmo KNN con $k = 5$ y 3 categorías

Otros casos de uso comunes incluyen sistemas de recomendación (a través de filtrado colaborativo) y detección de valores atípicos.
Notemos que aunque el algoritmo de KNN es muy simple y funciona con cualquier número de clases, además de que es fácil de agregar datos y tiene pocos parámetros tales como el valor de k y la matriz de distancias al nuevo punto a clasificar, el costo computacional es alto debido al cálculo de la distancia entre los puntos de datos para todas las muestras de entrenamiento. De igual modo, no es bueno con los datos dimensionales altos

1.8. Máquinas de Soporte Vectorial

1.8.1. Introducción

El método de clasificación-regresión Máquinas de Vector Soporte (Support Vector Machines, SVMs) fue desarrollado en la década de los 90, dentro de campo de la ciencia computacional. Si bien originariamente se desarrolló como un método de clasificación binaria, su aplicación se ha extendido a problemas de clasificación múltiple y regresión. SVMs ha resultado ser uno de los mejores clasificadores para un amplio abanico de situaciones, por lo que se considera uno de los referentes dentro del ámbito de aprendizaje estadístico y machine learning.
Las Máquinas de Vector Soporte se fundamentan en el Maximal Margin Classifier, que a su vez, se basa en el concepto de hiperplano. A lo largo de este ensayo se introducen por orden cada uno de estos conceptos. Comprender los fundamentos de las SVMs requiere de conocimientos sólidos en álgebra lineal. En este ensayo no se profundiza en el aspecto matemático, pero puede encontrarse una descripción detallada en el libro Support Vector Machines Succinctly by Alexandre Kowalczyk
En R, las librerías “e1071“ y “LiblineaR“ contienen los algoritmos necesarios para obtener modelos de clasificación simple, múltiple y regresión, basados en Support Vector Machines.

1.8.2. Hiperplano y Maximal Margin Classifier

En un espacio p-dimensional, un hiperplano se define como un subespacio plano y afín de dimensiones $p1$. El término afín significa que el subespacio no tiene por qué pasar por el origen. En un espacio de dos dimensiones, el hiperplano es un subespacio de 1 dimensión,

es decir, una recta. En un espacio tridimensional, un hiperplano es un subespacio de dos dimensiones, un plano convencional. Para dimensiones p>3 no es intuitivo visualizar un hiperplano, pero el concepto de subespacio con $p1$ dimensiones se mantiene.
La definición matemática de un hiperplano es bastante simple. En el caso de dos dimensiones, el hiperplano se describe acorde a la ecuación de una recta:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$$

Dados los parámetros β_0, β_1 y β_2, todos los pares de valores $x = (x_1, x_2)$ para los que se cumple la igualdad son puntos del hiperplano. Esta ecuación puede generalizarse para p-dimensiones:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p = 0$$

y de igual manera, todos los puntos definidos por el vector $(x = x_1, x_2, ..., x_p)$ que cumplen la ecuación pertenecen al hiperplano.
Cuando x no satisface la ecuación:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p < 0$$

o bien

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p > 0$$

el punto x cae a un lado o al otro del hiperplano. Así pues, se puede entender que un hiperplano divide un espacio p-dimensional en dos mitades. Para saber en qué lado del hiperplano se encuentra un determinado punto x, solo hay que calcular el signo de la ecuación.
La siguiente imagen muestra el hiperplano de un espacio bidimensional. La ecuación que describe el hiperplano (una recta) es $1+2x_1+3x_2 = 0$. La región azul representa el espacio en el que se encuentran todos los puntos para los que $1 + 2x_1 + 3x_2 > 0$ y la región roja el de los puntos para los que $1 + 2x_1 + 3x_2 < 0$.

1.8.3. Clasificación binaria empleando un hiperplano

Cuando se dispone de n observaciones, cada una con p predictores y cuya variable respuesta tiene dos niveles (de aquí en adelante identificados como +1 y 1), se pueden emplear hiperplanos para construir un clasificador que permita predecir a que grupo pertenece una observación en función de sus predictores. Este mismo problema puede abordarse también con otros métodos (regresión logística, LDA, árboles de clasificación…) cada uno con ventajas y desventajas.
Para facilitar la comprensión, las siguientes explicaciones se basan en un espacio de dos dimensiones, donde un hiperplano es una recta. Sin embargo, los mismos conceptos son aplicables a dimensiones superiores.
CASOS PERFECTAMENTE SEPARABLES LINEALMENTE
Si la distribución de las observaciones es tal que se pueden separar linealmente de forma perfecta en las dos clases (+1 y 1), entonces, un hiperplano de separación cumple que:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p > 0, \ si \ y_i = 1$$

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p < 0, \ si \ y_i = -1$$

Al identificar cada clase como +1 o 1, y dado que multiplicar dos valores negativos resultan en un valor positivo, las dos condiciones anteriores pueden simplificarse en una única:

$$y_i(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p) > 0, \quad \text{para} \ \ i = 1...n$$

Bajo este escenario, el clasificador más sencillo consiste en asignar cada observación a una clase dependiendo del lado del hiperplano en el que se encuentre. Es decir, la observación x^* se clasifica acorde al signo de la función $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + ... + \beta_p x_p^*$. Si $f(x^*)$ es positiva, la observación se asigna a la clase +1, si es negativa, a la clase 1. Además, la magnitud de $f(x^*)$ permite saber cómo de lejos está la observación del hiperplano y con ello la confianza de la clasificación.
La definición de hiperplano para casos perfectamente separables linealmente resulta en un número infinito de posibles hiperplanos, lo que hace necesario un método que permita seleccionar uno de ellos como clasificador óptimo.
La solución a este problema consiste en seleccionar como clasificador óptimo al que se conoce como *maximal margin hyperplane o hiperplano óptimo de separación*, que se corresponde con el hiperplano que se encuentra más alejado de todas las observaciones de entrenamiento. Para obtenerlo, se tiene que calcular la distancia perpendicular de cada observación a un determinado hiperplano. La menor de estas distancias (conocida como margen) determina como de alejado está el hiperplano de las observaciones de entrenamiento. El maximal margin hyperplane se define como el hiperplano que consigue un mayor margen, es decir, que la distancia mínima entre el hiperplano y las observaciones es lo más grande posible. Aunque esta idea suena razonable, no es posible aplicarla, ya que habría infinitos hiperplanos contra los que medir las distancias. En su lugar, se recurre a métodos de optimización. Para encontrar una descripción más detallada de la solución por optimización consultar (*Support Vector Machines Succinctly by Alexandre Kowalczyk*).
La imagen anterior muestra el maximal margin hyperplane para un conjunto de datos de entrenamiento. Las tres observaciones equidistantes respecto al maximal margin hyperplane se encuentran a lo largo de las líneas discontinuas que indican la anchura del margen. A estas observaciones se les conoce como vectores soporte, ya que son vectores en un espacio p-dimensional y soportan (definen) el maximal margin hyperplane. Cualquier modificación en estas observaciones (vectores soporte) conlleva cambios en el maximal margin hyperplane. Sin embargo, modificaciones en observaciones que no son vector soporte no tienen impacto alguno en el hiperplano.
CASOS CUASI-SEPARABLES LINEALMENTE
El maximal margin hyperplane descrito en el apartado anterior es una forma muy simple y natural de clasificación siempre y cuando exista un hiperplano de separación. En la gran mayoría de casos reales, los datos no se pueden separar linealmente de forma perfecta, por lo que no existe un hiperplano de separación y no puede obtenerse un maximal margin hyperplane.
Para solucionar estas situaciones, se puede extender el concepto de *maximal margin hyperplane* para obtener un hiperplano que casi separe las clases, pero permitiendo que cometa unos pocos errores. A este tipo de hiperplano se le conoce como *Support Vector Classifier o Soft Margin*

1.8.4. Support Vector Classifier o Soft Margin SVM

El Maximal Margin Classifier descrito en la sección anterior tiene poca aplicación práctica, ya que rara vez se encuentran casos en los que las clases sean perfecta y linealmente separables. De hecho, incluso cumpliéndose estas condiciones ideales, en las que exista un hiperplano capaz de separar perfectamente las observaciones en dos clases, esta aproximación sigue presentando dos inconvenientes:
Dado que el hiperplano tiene que separar perfectamente las observaciones, es muy sensible a variaciones en los datos. Incluir una nueva observación puede suponer cambios muy grandes en el hiperplano de separación (poca robustez).

Que el maximal margin hyperplane se ajuste perfectamente a las observaciones de entrenamiento para separarlas todas correctamente suele conllevar problemas de overfitting.
Por estas razones, es preferible crear un clasificador basado en un hiperplano que, aunque no separe perfectamente las dos clases, sea más robusto y tenga mayor capacidad predictiva al aplicarlo a nuevas observaciones (menos problemas de overfitting). Esto es exactamente lo que consiguen los clasificadores de vector soporte, también conocidos como soft margin classifiers o Support Vector Classifiers. Para lograrlo, en lugar de buscar el margen de clasificación más ancho posible que consigue que las observaciones estén en el lado correcto del margen; se permite que ciertas observaciones estén en el lado incorrecto del margen o incluso del hiperplano.
La siguiente imagen muestra un clasificador de vector soporte ajustado a un pequeño set de observaciones. La línea continua representa el hiperplano y las líneas discontinuas el margen a cada lado. Las observaciones 2, 3, 4, 5, 6, 7 y 10 se encuentran en el lado correcto del margen (también del hiperplano) por lo que están bien clasificadas. Las observaciones 1 y 8, a pesar de que se encuentran dentro del margen, están en el lado correcto del hiperplano, por lo que también están bien clasificadas. Las observaciones 11 y 12, se encuentran en el lado erróneo del hiperplano, su clasificación es incorrecta. Todas aquellas observaciones que, estando dentro o fuera del margen, se encuentren en el lado incorrecto del hiperplano, se corresponden con observaciones de entrenamiento mal clasificadas.
La identificación del hiperplano de un clasificador de vector soporte, que clasifique correctamente la mayoría de las observaciones a excepción de unas pocas, es un problema de optimización convexa. Si bien la demostración matemática queda fuera del objetivo de esta introducción, es importante mencionar que el proceso incluye un hiperparámetro de tuning C. C controla el número y severidad de las violaciones del margen (y del hiperplano) que se toleran en el proceso de ajuste. Si $C = \infty$, no se permite ninguna violación del margen y por lo tanto, el resultado es equivalente al Maximal Margin Classifier (teniendo en cuenta que esta solución solo es posible si las clases son perfectamente separables). Cuando más se aproxima C a cero, menos se penalizan los errores y más observaciones pueden estar en el lado incorrecto del margen o incluso del hiperplano. C es a fin de cuentas el hiperparámetro encargado de controlar el balance entre bias y varianza del modelo. En la práctica, su valor óptimo se identifica mediante cross-validation.
El proceso de optimización tiene la peculiaridad de que solo las observaciones que se encuentran justo en el margen o que lo violan influyen sobre el hiperplano. A estas observaciones se les conoce como vectores soporte y son las que definen el clasificador obtenido. Esta es la razón por la que el parámetro C controla el balance entre bias y varianza. Cuando el valor de C es pequeño, el margen es más ancho, y más observaciones violan el margen, convirtiéndose en vectores soporte. El hiperplano está, por lo tanto, sustentado por más observaciones, lo que aumenta el bias pero reduce la varianza. Cuando mayor es el valor de C, menor el margen, menos observaciones serán vectores soporte y el clasificador resultante tendrá menor bias pero mayor varianza.
Otra propiedad importante que deriva de que el hiperplano dependa únicamente de una pequeña proporción de observaciones (vectores soporte), es su robustez frente a observaciones muy alejadas del hiperplano. Esto hace al método de clasificación vector soporte distinto a otros métodos tales como Linear Discrimiant Analysis (LDA), donde la regla de clasificación depende de la media de todas las observaciones.