FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

Métodos de clasificación estadística

Pablo Estévez Domarco

2020/2021

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Métodos de clasificación estadística

Pablo Estévez Domarco

09/2021

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Trabajo propuesto

Área de Conocimiento: Estadística e investigación operativa
Título: Métodos de clasificación estadística
Breve descrición del contenido
Exposición de diversos métodos para la clasificación estadística.
Observaciones
Para la puesta en práctica de los métodos se usó el lenguaje R.

Índice general

Resumen

En esta memoria se abordarán diversos métodos para la clasificación estadística. Esto son, modelos de predicción para una variable de salida categórica. Además de la explicación formal y teórica de cada método, se aplicarán los modelos descritos sobre datos reales obtenidos a través de distintos repositorios en internet. Finalmente se validarán los modelos y se extraerán conclusiones acerca de la efectividad de cada método para las distintas naturalezas de los datos tratados.

Abstract

Various methods for statistical classification will be addressed in this report. That is, prediction models for a categorical output variable. In addition to the formal and theoretical explanation of each method, the models described will be applied to real data obtained through different repositories on the internet. Finally, the models will be validated and conclusions will be drawn about the effectiveness of each method for the different natures of the data processed.

Capítulo 1

El problema de la clasificación estadística

En el campo de la Estadística, uno de los problemas más importantes es el de la clasificación. Dado un conjunto de datos organizados en distintas categorías o clases, se pretende identificar a cuál de ellas pertenece una nueva observación. El fondo del problema reside en reconocer patrones y tendencias en los datos, de tal forma que se puedan realizar predicciones fidedignas. Aunque pueda parecer algo abstracto, en la realidad existen numerosos ejemplos en diversas áreas del conocimiento que van desde la ciencia o medicina hasta la industria o las finanzas. Algunos ejemplos podrían ser: identificar si un paciente está sano o enfermo a partir de un test que mida diversas variables fisiológicas, asignar un correo electrónico en la categoría de "spam", implementar un procedimiento automático que reconozca letras y números escritos a mano, determinar la especie de una planta a partir de diversos rasgos como la longitud de los pétalos, etc.

Aunque tratan problemáticas completamente distintas, estos ejemplos tienen ciertos elementos en común. En todos ellos, para cada observación existe un conjunto de parámetros preestablecidos (cantidad de cierta hormona en sangre, número de veces que se apela al destinatario, valor en la escala de grises de los píxeles en una imagen...) que influyen en su categoría (enfermo/sano, *spam*/no *spam*, letra "A", especie *Iris setosa*...).

1.1. La clasificación supervisada

Desde el punto de vista matemático, el objetivo de la clasificación estadística es construir una *función de decisión* f, llamada también *clasificador* o *función de predicción*, que asigne una categoría a cada observación. Si para realizar esta predicción se consideran p parámetros (también denominados predictores), la *variable de entrada* (o *input*), X, será

un vector de dimensión p cuyos componentes son los valores que toman estos parámetros y se denotarán en minúscula, $X^T = (x_1, x_2, \ldots, x_p)$. Por otro lado, la *variable de salida* (o *output*), G, es categórica ya que toma el valor de la clase a la que pertenece cada observación X dada. El conjunto de salida, formado por las K categorías donde se pretende clasificar, se denotará como $\mathcal{G}$. En adelante, cada clase se codificará con un número natural, por lo que $G = \{1, \ldots, K\}$. En definitiva, f es una función de $\mathbb{R}^p$ en $\mathcal{G} = \{1, \ldots, K\}$. En referencia a la notación, a la función de predicción, hasta ahora denotada como f, comúnmente se le denomina $\hat{G}$. De tal forma que si se escribe $\hat{G}(X) = 1$, significa que la predicción para el dato X es que pertenece a la clase codificada como 1.

La función de decisión se puede hallar a través de diversos métodos, los cuales serán más o menos adecuados en función de las características del problema a tratar. Una posibilidad es usar el teorema de Bayes y las probabilidades condicionadas. En el contexto de la clasificación, el teorema de Bayes se puede expresar en los siguientes términos

$$P(G = g|X) = \frac{P(X|G = g)P(G = g)}{P(X)} \tag{1.1}$$

donde $P(G = g|X)$ es la probabilidad a posteriori de que la observación X pertenezca a la clase g, $P(X|G = g)$ es la probabilidad del dato X considerando que pertenece a la clase g, $P(G = g)$ y $P(X)$ son las probabilidades a priori. La función de predicción se elegiría de tal manera que clasifique las observaciones en la categoría que maximice esta probabilidad a posteriori. Esto no es en absoluto trivial pues en la práctica no se conocen las distribuciones de los datos que se pretenden clasificar.

Existen dos grandes familias en la clasificación estadística: la *supervisada* y la *no supervisada*. En la supervisada se tiene un conocimiento a priori, un conjunto de datos, denominado *conjunto de entrenamiento*, de los que se sabe la clase a la que pertenecen. A partir de esta muestra se elabora la función de predicción, $\hat{G}$, que sea capaz de clasificar nuevas observaciones. Esto es, para un conjunto de entrenamiento de N observaciones, $\{X_1, X_2, \ldots, X_N\}$ se conocen sus clases $\{G_1, G_2, \ldots, G_N\}$. La función de predicción se definirá minimizando una función de pérdida $L(\hat{G}(X_i), G_i)$ que penalizará las equivocaciones de la regla de decisión en la muestra de entrenamiento.

Por el contrario, en la clasificación no supervisada no se dispone de este conocimiento a priori, no se tiene un conjunto de datos ya clasificados. De hecho, ni se tienen por qué conocer las posibles clases. En esta memoria, se abordarán distintos métodos enfocados a resolver los problemas del primer tipo.

1.2. El equilibro en las muestras

Como se expuso anteriormente, en el ámbito de la clasificación supervisada se cuenta con un conjunto de entrenamiento de datos ya clasificados. Cuando para cada una de las clases se tiene un número parecido de datos, el problema se denomina *equilibrado*. Esto es lo ideal, pues ninguna categoría está sobrerrepresentada con respecto a otras y no se tenderá a sesgar la clasificación en favor de las categorías con más datos. Sin embargo, puede suceder que en la muestra de entrenamiento exista una clase minoritaria, es decir, con muchas menos muestras que el resto. Esto es muy común en el ámbito sanitario por ejemplo, donde las muestras de pacientes sanos son más numerosas que las de enfermos. Este desequilibrio en el conjunto de entrenamiento perjudica a las clases minoritarias. Volviendo al caso anterior, si se tiene una muestra de 100 pacientes de los cuáles 95 están sanos y 5 enfermos, un método de predicción que clasifique a todos los pacientes como sanos tendría un porcentaje de acierto del 95 % dando una falsa apariencia de fiabilidad, si bien lo más conveniente en este contexto sería que se clasifique a un sano como enfermo antes que a un enfermo como sano.

Para solventar este problema, se estudiarán metodologías específicas para cada método en particular. Aun así, se pueden considerar ciertas pautas en común.

- **Función de pesos.** Una posible solución sería construir una función que pondere los datos y de más significación a aquellos pertenecientes a las clases minoritarias. Cómo crear esta función es un problema abierto.

- **Equilibrar las muestras.** Dado que las clases están en desigualdad de condiciones, se podría actuar en dos sentidos: reducir la muestra que se toma de las clases mayoritarias o bien aumentar los datos de las categorías con menos muestras. Sea como fuere, habría que prestar especial atención a nuevas problemáticas que podrían surgir. Disminuir el número de datos podría incrementar la varianza de la regla de decisión, haciéndola menos fiables. En tanto que aumentar las muestras implicaría simular datos artificialmente asumiendo ciertos parámetros e incluso distribuciones, lo cuál es arriesgado puesto que las nuevas muestras generadas podrían alejarse de la realidad.

1.3. La clasificación binaria

El caso más sencillo de clasificación es la *binaria*. En estos problemas, solamente se tienen dos categorías: "éxito" o "fracaso", codificadas con un dígito que tomará el valor 0 ó 1 por lo que el conjunto de salida es $\mathcal{G} = \{0, 1\}$. Por ejemplo, en el caso del test médico

donde las categorías son "sano" y "enfermo" se podría codificar a los pacientes sanos como $G = 0$ y a los enfermos como $G = 1$.

En los problemas binarios, un posible enfoque sería emplear un modelo de predicción con variable de salida Y cuantitativa. Después, se elegiría una función de decisión por la que a cada variable de salida se le asignaría una categoría en función del valor de la predicción $\hat{Y}$. Un ejemplo sería restringir los valores de $\hat{Y}$ al intervalo $[0, 1]$ y hacer la predicción de clase mediante la regla

$$\hat{G} = \begin{cases} 0 & si \quad \hat{Y} < 0.5 \\ \\ 1 & si \quad \hat{Y} \geq 0.5 \end{cases} \tag{1.2}$$

El valor de $\hat{Y}$ a partir del cuál se clasifica en una categoría u en otra, en este caso 0.5, se conoce como *threshold*. A pesar de que pareciera lógico tomarlo siempre como 0.5, puede ser conveniente modificar este valor. Por ejemplo si el modelo presenta un porcentaje alto de falsos negativos o positivos.

A pesar de que no todos los problemas que se surgen en el ámbito de la clasificación estadística son de este tipo, la clasificación binaria es un caso muy frecuente y de suma importancia dado que la respuesta puede asociarse con la distribución de *Bernoulli* (éxito o fracaso). Si se quiere extender una regla de decisión binaria a más de dos categorías, sean K, se podrían emplear alguna de las siguientes dos estrategias:

- ***Uno contra el resto.*** Este método consiste en usar el modelo binario considerando por un lado una clase y por otro el resto juntas. Después de K iteraciones, una nueva observación se clasificará en aquella categoría que haya obtenido mejores resultados en el cómputo global. El principal inconveniente de este método es que podría degenerar en un problema no equilibrado, puesto que la clase que engloba a todas las categorías menos una, tenderá a tener más datos. Pese a esto, es interesante considerar este procedimiento ya que computacionalmente es eficiente, solo hay que repetir el modelo binario K veces.

- ***Uno contra uno.*** Se organizan emparejamientos que enfrenten a toda las clases entre sí. En cada uno, se aplica el modelo binario y a la categoría ganadora se le suma un punto. Cuando todas las clases se hayan comparado entre sí, la categoría que más puntos tenga será la que se le asigne a la nueva observación. Este método resuelve el problema del equilibrio en comparación con el *Uno contra el resto*, supuesto el equilibrio inicial entre las clases. El inconveniente es que es mucho menos eficiente, puesto que habría que realizar $\binom{K}{2}$ iteraciones.

1.4. Validación de los métodos

Una vez formulados los distintos métodos de clasificación en un conjunto de datos es importante validar su eficacia, es decir, valorar en que medida clasificarán correctamente las nuevas observaciones.

En esta memoria se usará la *validación cruzada* para evitar los problemas de sobreajuste, es decir, que el modelo se adapte bien al la muestra de entrenamiento pero que no se ajuste adecuadamente para predecir un nuevo conjunto de datos.

Esta estrategia consiste en dividir aleatoriamente la muestra de entrenamiento en dos partes no necesariamente iguales (en este trabajo se tomará el 80 % de los datos para una y el 20 % para la otra). La parte de mayor tamaño, denominada conjunto de entrenamiento, se usa para ajustar el modelo, es decir, obtener los distintos parámetros que definen la función de clasificación.

Por otro lado, el conjunto de datos restante, o conjunto de validación, se usa en el modelo para obtener las clases estimadas. Después, se comparan las clases reales de estos datos con las predichas por el modelo y se calcula el error.

Capítulo 2

Regresión logística

2.1. El modelo de regresión logística

La *regresión logística* es un método de clasificación binario que se basa en el uso de la regresión lineal para calcular la probabilidad a posteriori de que una observación pertenezca a una clase $P(G = g|X)$ (veáse 1.1).

Es importante tener en cuenta que al haber solo dos categorías, sean $\mathcal{G} = \{0, 1\}$, estas probabilidades son complementarias: $P(G = 0|X) + P(G = 1|X) = 1$. Además, la probabilidad condicionada de $G = 1$ es precisamente la esperanza condicionada, en adelante μ

$$\mu = E(G|X) = P(G = 1|X) \tag{2.1}$$

Supóngase un problema de clasificación binario donde la variable de entrada X es, por ahora, escalar (i.e, $p = 1$). Supóngase también un conjunto de entrenamiento $\{X_1, X_2, \ldots, X_N\}$ con clases $\{G_1, G_2, \ldots, G_N\}$. Un posible planteamiento sería usar un modelo de regresión lineal por mínimos cuadrados del siguiente modo.

Se modela el valor de una variable salida cuantitativa Y según la recta $Y = \beta_0 + \beta_1 X$, donde β_0 y β_1 son los llamados *coeficientes de regresión*, elegidos por el método de mínimos cuadrados. En resumen, esta elección de β_0 y β_1 (que se denota $\hat{\beta}_0$, $\hat{\beta}_1$) se hace de tal forma que minimice la suma de residuos al cuadrado, RSS, que en este caso sería

$$RSS(\beta_0, \beta_1) = \sum_{i=1}^{N} (G_i - \beta_0 - \beta_1 X_i)^2 \tag{2.2}$$

La intención de considerar la variable cuantitativa Y es que sirva de modelo para el cálculo de la probabilidad $P(G = g|X)$. El problema es que al tratarse de una recta no hay seguridad de que las predicciones, $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$, tomen valores en el intervalo $[0, 1]$.

En el siguiente ejemplo se pretende clasificar billetes como auténticos, $G = 0$, o falsos, $G = 1$. Para ello, se usa una cámara industrial que digitaliza cada billete y una función matemática, *Wavelet Transform* (WT), que proporciona cuatro parámetros para cada imagen: varianza, asimetría, curtosis y entropía.

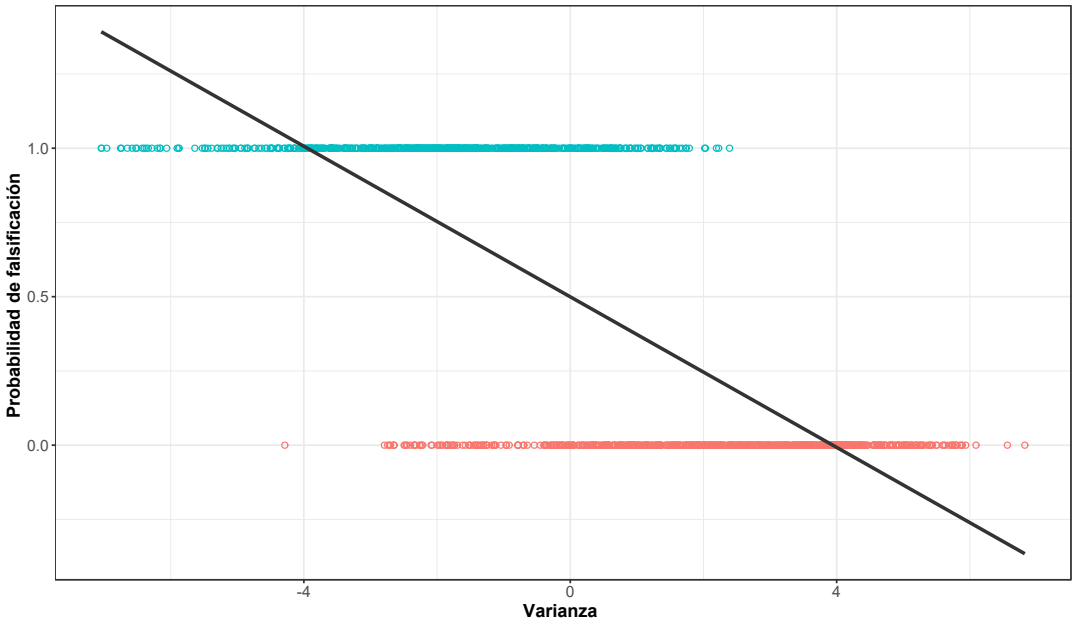

Figura 2.1: Modelo de regresión lineal por mínimos cuadrados que pretende modelar la probabilidad de falsificación en función de la varianza que da WT.

Como se puede observar en la Figura 2.1, usando la regresión lineal se predicen probabilidades mayores que 1 para varianzas bajas y menores que 0 para varianzas altas.

Para corregir esto, en la regresión logística se usa una función que restrinja el valor de las probabilidades al intervalo $[0, 1]$. Una de las posibles funciones que se pueden usar con tal fin es la llamada *función sigmoide*, σ

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

Nótese que $\lim_{x \to +\infty} \sigma(x) = 1$ y $\lim_{x \to -\infty} \sigma(x) = 0$.

Usando entonces un modelo lineal $\beta_0 + \beta_1 X$ en (2.3), se obtiene un modelo para hallar la probabilidad a posteriori de que un determinado dato pertenezca a una clase. Sin pérdida de generalidad, considérese esta categoría la codificada como $G = 1$

$$P(G = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}} = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} \tag{2.4}$$

El valor de los coeficientes β_0 y β_1 se estimará por máxima verosimilitud. Este ajuste se explicará más adelante.

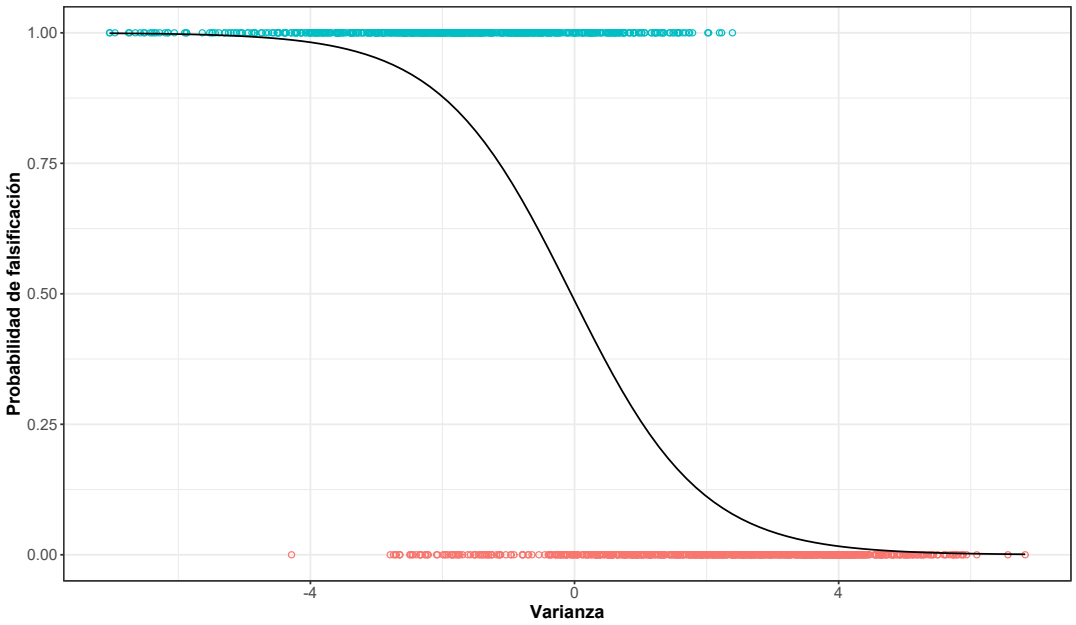

Figura 2.2: Modelo de regresión logística. Ahora si que las probabilidades si que pertenecen al intervalo [0,1].

Como se comentó previamente, al tratarse de un problema binario la probabilidad de pertenecer a la otra clase, $G = 0$, es la complementaria

$$P(G = 0|X) = 1 - P(G = 1|X) = \frac{1}{1 + e^{\beta_0 + \beta_1 X}} \tag{2.5}$$

Si se dividen ambas probabilidades y se toman logaritmos se llega a la función denominada *logit* o *log-odds*

$$\log\left(\frac{P(G = 1|X)}{P(G = 0|X)}\right) = \beta_0 + \beta_1 X \tag{2.6}$$

La finalidad de considerar esta función es facilitar el ajuste del modelo, que ahora sí puede ser una recta ya que no se está modelando la probabilidad directamente, sino el logaritmo de las *odds*.

Considérese un ensayo de *Bernoulli*, es decir, un experimento aleatorio con únicos resultados *éxito* con probabilidad p, o *fracaso* con probabilidad $q = 1 - p$. Las *odds* o *razón de probabilidad* de éxito, es la razón $\frac{p}{q}$. Si, por ejemplo, se tiene que la probabilidad de éxito es $p = 0.4$, entonces las *odds* de éxito son $0.4/0.6 = 2/3$. Esto equivale a decir que se esperan 2 éxitos por cada 3 fracasos.

Esta transformación de probabilidades a *odds* es monótona, si la probabilidad crece las *odds* también lo harán. Además, el rango de valores ya no se limita al intervalo $[0, 1]$, puesto que las *odds* toman valores en $[0, \infty]$. Si además se toman logaritmos, finalmente se llega a la función *logit*, que toma valores en $[-\infty, +\infty]$, por lo que ahora si tiene sentido considerar un modelo lineal.

Como se comentó anteriormente, se pueden utilizar otras funciones además de la *logit* para pasar del intervalo $[0, 1]$ de las probabilidades a toda la recta real. En la terminología de la clasificación estadística, la función usada con tal fin se denomina función *link* y se suele denotar g. Por tanto, la función que modela las probabilidades es la inversa de la función *link*, g^{-1}. En el caso de usar la función *logit*, $g^{-1}(x) = \sigma(x)$, como se muestra en (2.4).

Una vez estimados los coeficientes β_0 y β_1, se clasificaría cada observación en función del valor que tome la probabilidad $P(G|X)$ según el modelo (2.4). Para ello, se establece un *threshold*, es decir, se clasifica la observación X en la clase $G = 1$ si $P(G = 1|X) > 0.5$, por ejemplo, y en la clase $G = 0$ en el caso contrario.

En el caso multivariante, es decir cuando las observaciones son vectores $X^T = (x_1, x_2, \ldots, x_p)$ con $p > 1$, los principios de la regresión logística son los mismos. La diferencia es que ahora se pasará de tener dos coeficientes, β_0 y β_1, a tener $p + 1$

$$\log\left(\frac{P(G = 1|X)}{P(G = 0|X)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \tag{2.7}$$

Para facilitar la notación, se incluirá la constante 1 como el primer componente da cada vector de entrada X. De esta manera, se podrá incluir el intercepto β_0 en el vector $\beta^T = (\beta_0, \beta_1, \ldots, \beta_p)$. Por tanto, la anterior ecuación se escribirá

$$\log\left(\frac{P(G = 1|X)}{P(G = 0|X)}\right) = \beta^T X \tag{2.8}$$

El valor de la probabilidad a posteriori será

$$P(G = 1|X) = \frac{e^{\beta^T X}}{1 + e^{\beta^T X}} \tag{2.9}$$

2.2. Ajuste del modelo por máxima verosimilitud

Supuesto el modelo, el problema ahora reside en la estimación del parámetro $\beta^T = (\beta_0, \beta_1, \ldots, \beta_p)$. Esta estimación se denotará $\hat{\beta}^T = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ y se realizará por máxima verosimilitud, esto es, de tal forma que se maximice la probabilidad de clasificar las observaciones de la muestra de entrenamiento $\{X_1, X_2, \ldots, X_N\}$ en sus respectivas clases $\{G_1, G_2, \ldots, G_N\}$.

La función de verosimilitud derivada de suponer que la variable respuesta sigue una distribución binomial, extensión de la *Bernoulli*, es

$$\mathcal{L}(\beta) = \prod_{i=1}^{N} P(G = 1|X_i)^{G_i}(1 - P(G = 1|X_i))^{1-G_i} \tag{2.10}$$

Tomando logaritmos se obtiene la función logarítmica de verosimilitud, $l = \log \mathcal{L}$. Desarrollando se llega a

$$\begin{aligned} l(\beta) &= \sum_{i=1}^{N} \log P(G = 1|X_i)^{G_i} + \sum_{i=1}^{N} \log(1 - P(G = 1|X_i))^{1-G_i} = \\ &= \sum_{i=1}^{N} \{G_i \log P(G = 1|X_i) + (1 - G_i) \log(1 - P(G = 1|X_i))\} \end{aligned} \tag{2.11}$$

Finalmente, usando la notación vectorial de (2.13) se tiene que

$$l(\beta) = \sum_{i=1}^{N} \left\{ G_i \beta^T X_i - \log(1 + e^{\beta^T X_i}) \right\} \tag{2.12}$$

El estimador de máxima verosimilitud $\hat{\beta}^T = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ será aquel que maximice esta función (véase [1, Capítulo 4]). La función de estimación de las probabilidades a posteriori será por tanto

$$\hat{P}(G = 1|X) = \frac{e^{\hat{\beta}^T X}}{1 + e^{\hat{\beta}^T X}} \tag{2.13}$$

2.3. Caso práctico: billetes falsos

Consideremos el ejemplo expuesto anteriormente donde se clasifican billetes como auténticos, $G = 0$, o falsos, $G = 1$. A partir de las imágenes digitalizadas de los billetes, la función WT devuelve valores de cuatro parámetros: varianza, asimetría, curtosis y entropía. Si se consideran todos estos parámetros en la función de decisión se tiene entonces que $p = 4$. La variable de entrada en este caso es un vector $X = (1, x_1, \ldots, x_p)$.

Para ajustar el modelo se toma una muestra de entrenamiento con el 80 % de los datos originales, es decir, 1098 billetes. El conjunto de validación está formado por el restante 20 %, esto es, 274 muestras. Los resultados del modelo de regresión logística ajustado por máxima verosimilitud se muestran en el Tabla 2.2.

El dato del *p-valor* que se muestra en las tablas da una idea de la significación de cada predictor. En este caso, la contribución de la variable *entropía* resulta no significativa

Tabla 2.1: Muestra aleatoria de cuatro datos del conjunto de entrenamiento. En este ejemplo, se tiene un total de N=1372 billetes clasificados, de los cuales 610 son falsificaciones y 762 son auténticos. Se puede considerar, por tanto, que la muestra de estrenamiento está equilibrada.

	varianza	asimetria	curtosis	entropia	clase
#5	0.32924	-4.45520	4.571800	-0.98880	0
#75	4.40690	10.90720	-4.577500	-4.42710	0
#825	-0.42940	-0.14693	0.044265	-0.15605	1
#987	0.84546	3.48260	-3.630700	-1.39610	1

Tabla 2.2: Estimación de los coeficientes, error estándar y *p-valor* de cada predictor.

	Coeficientes	Error estándar	p-valor
(Intercept)	7.321805	1.5588603	0.0000026
varianza	-7.859331	1.7383123	0.0000061
asimetria	-4.190963	0.9041488	0.0000036
curtosis	-5.287431	1.1611830	0.0000053
entropia	-0.605319	0.3307210	0.0672050

para el modelo. Esto no quiere decir que por si solo este predictor no sea significativo, no obstante en presencia del resto de variables ya no es necesario.

Excluyendo entonces este parámetro del modelo se obtienen los resultados mostrados en el Tabla 2.3.

Se tiene pues la función $\hat{P}(G = 1|X)$ que predice la probabilidad de falsificación a partir de la variable de entrada $X = (1, x_1, x_2, x_3)$ cuyos componentes son, respectivamente, los parámetros de varianza, asimetría y curtosis obtenidos a través de la digitalización de cada billete

$$P(Billete\ Falso) = \hat{P}(G = 1|X) = \frac{e^{\hat{\beta}^T X}}{1 + e^{\hat{\beta}^T X}} \tag{2.14}$$

donde $\hat{\beta}^T \approx (6.89, -6.78, -3.51, -4.46)$ (véase Tabla 2.3).

Una vez determinada el modelo de predicción para las probabilidades a posteriori (2.14), el objetivo ahora es clasificar los billetes, es decir, definir la función de decisión $\hat{G}$. Para

Tabla 2.3: Resultados obtenidos al excluir la *entropía*.

	Coeficientes	Error estándar	p-valor
(Intercept)	6.884972	1.3838479	7.0e-07
varianza	-6.783457	1.3949643	1.2e-06
asimetria	-3.506680	0.6932163	4.0e-07
curtosis	-4.464192	0.9006030	7.0e-07

este ejemplo se considerará un *threshold* de 0.5, luego dado un billete X

$$\hat{G}(X) = \begin{cases} 0 \quad si \quad \hat{P}(G=1|X) \leq 0.5 \\ \\ 1 \quad si \quad \hat{P}(G=1|X) > 0.5 \end{cases} \tag{2.15}$$

Para evaluar el modelo se comparan las clases reales del conjunto de validación con las predicciones obtenidas por la función de clasificación. De esta forma se construye una matriz de confusión $(m_{ij}) \in \mathcal{M}_{2\times 2}$, donde el elemento m_{ij} indica el número de datos con clase $G = j - 1$ clasificados como $G = i - 1$ (Figura 2.3).

		Clase Real	
		Auténtico	Falsificación
Predicción	Auténtico	148	1
	Falsificación	4	121

Figura 2.3: Matriz de confusión para el conjunto de validación.

El modelo clasifica correctamente el $\dfrac{148+121}{274} = 98.17\,\%$ de las observaciones del conjunto de validación.

La interpretación de los coeficientes de $\hat{\beta}$ es la siguiente. Para el caso de la *varianza* ($\beta_1 \approx -6.78$), el *log-odds* de que un billete sea falso está negativamente relacionado con este parámetro. En concreto, por cada unidad que se incrementa la *varianza*, el *log-odds* de falsificación disminuye 6.78 unidades.

Capítulo 3

Modelos aditivos generalizados

Como se expuso en el capítulo anterior, un planteamiento para el problema de la clasificación binaria es definir una función para estimar las probabilidades a posteriori $P(G = 1|X)$ o, equivalentemente, μ (véase 2.1). Para ello, es posible emplear un modelo

$$Y = f(X) + \epsilon \tag{3.1}$$

donde ϵ denota el error de la predicción f respecto al valor real Y. La variable de salida no está restringida necesariamente al intervalo $[0, 1]$ gracias al uso de una función *link* g. La esperanza condicionada ya no está directamente relacionada con la variable de entrada X, sino que lo hace a través de esta función

$$\mu = E(G|X) = g^{-1}(Y) \tag{3.2}$$

Esto es precisamente un fundamento de los denominados *Modelos Lineales Generalizados* (GLM). En los GLM además de usar una función *link*, se asume que la función f, que predice la variable de salida Y, es una combinación lineal de los componentes de la variable de entrada

$$f(X) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \tag{3.3}$$

El modelo lineal por mínimos cuadrados o la regresión logística son casos particulares de GLM, donde la función *link* es la identidad o la *logit*, respectivamente.

Esta suposición de linealidad facilita la determinación de la función f ya que en lugar de tener que estimar una función p-dimensional $f(x_1, x_2, \ldots, x_p)$, se estiman los $p + 1$ coeficientes β. El uso de una función *link* permite superar la limitación del modelo lineal de que la variable de salida Y se distribuya de manera normal. Los GLM se pueden emplear para variables de salida que sigan distribuciones de la familia exponencial: Normal, Gamma, Bernoulli, Poisson...

A pesar de las mejoras que aportan, los GLM sigue presentado una serie de inconvenientes. La principal limitación de estos modelos es la presunción de una relación de linealidad entre las variables de salida y entrada. Este planteamiento simplifica el ajuste del modelo y conlleva interpretaciones de más fácil comprensión. No obstante, la suposición de linealidad no es una buena aproximación para una gran parte de casos en la práctica.

En este contexto se plantea el *Modelo Aditivo Generalizado* (GAM) con el fin de superar la linealidad. Este modelo es una extensión del GLM donde las relación entre los predictores y la variable salida deja de ser lineal si bien se mantiene la aditividad.

3.1. Expansión lineal de bases y regresión polinómica

En los GAM se modela la variable salida Y con una *expansión lineal de bases*

$$f(X) = \sum_{m=0}^{M} \beta_m h_m(X) \tag{3.4}$$

Donde $h_m(X) : \mathbb{R}^p \mapsto \mathbb{R}$ denota una transformación de X no necesariamente lineal (véase [7]). Esto es, en estos modelos se remplaza la variable de entrada X por M transformaciones. El punto crucial de esta aproximación es que una vez determinada la *base de funciones*, i.e las h_m, el modelo es lineal en las nuevas variables determinadas por estas transformaciones. Por tanto, se facilita el ajuste pudiendo usarse los mismos métodos que en los modelos lineales.

Nótese que el GAM sigue siendo una generalización del modelo lineal que se obtiene tomando $h_m(X) = x_m$ con $m = 1, \ldots, p$. El uso de una base de funciones incrementa las posibilidades para llegar a una estimación más fiable de la variable de salida Y y, por consiguiente, de la probabilidad a posteriori para clasificar las observaciones.

Con la finalidad de simplificar los resultados y aclarar los conceptos básicos, en adelante se considerará el caso univariante, es decir, la variable de entrada X es un escalar, por lo que las transformaciones h_m son funciones de $\mathbb{R}$ en $\mathbb{R}$.

La base polinómica consiste en modelar Y como un polinomio de grado M con $h_m(X) = X^M$. Se tiene entonces

$$f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_M X^M \tag{3.5}$$

Esto se conoce como *regresión polinómica*. Los coeficientes β_i con $i = 1, \ldots, M$ se pueden ajustar a través de mínimos cuadrados, ya que el modelo se puede considerar lineal con variables de entrada $X, X^2, \ldots, X^M$.

En consecuencia, para un problema de clasificación binaria se puede emplear la regresión polinómica junto con la función *logit* como *link* (Figura 3.1). El modelo para la probabilidad

a posteriori será

$$P(G=1|X) = \frac{e^{\beta_0+\beta_1 X+\beta_2 X^2+\cdots+\beta_M X^M}}{1+e^{\beta_0+\beta_1 X+\beta_2 X^2+\cdots+\beta_M X^M}} \tag{3.6}$$

A medida que aumenta el grado del polinomio la función presenta cada vez una mayor "flexibilidad" y se comporta de forma extraña, particularmente en los los extremos. Por este motivo, en la práctica es aconsejable usar polinomios de grado no mayor que 4 (véase [7]).

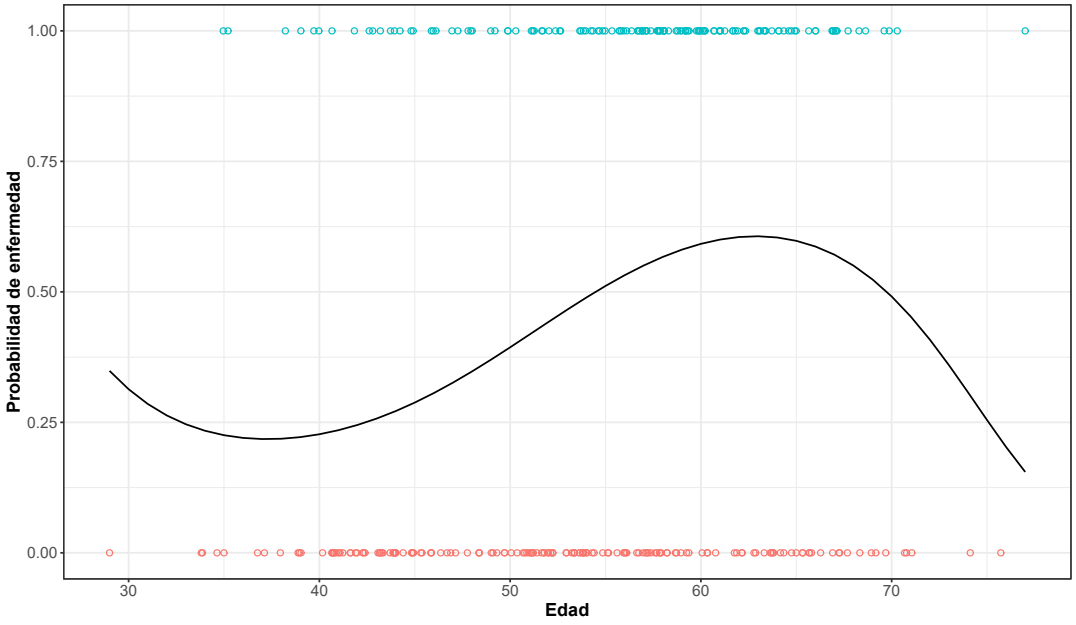

Figura 3.1: Ejemplo de modelo de regresión polinómica logística de grado 3. El modelo se obtuvo a través de datos médicos de diversos pacientes que presentan una enfermedad cardíaca, G=1, o que están sanos, G=0. En la figura se modela la probabilidad de padecer una enfermedad en función de la edad.

3.2. Polinomios por partes

El método de regresión polinómica se puede perfeccionar dividiendo el dominio de X en intervalos contiguos y definiendo f por partes, es decir, como un polinomio distinto para cada intervalo. Esto se denomina polinomio por partes o en inlgés *piecewise polynomial*. De esta manera, se pueden conseguir modelos más acordes a la realidad.

Si se divide el dominio de la variable de entrada a través de K puntos de corte $\{\xi_1, \ldots, \xi_K\}$, en adelante *nodos*, se obtendrán $K+1$ intervalos. Por tanto, f estará definida en $K+1$ polinomios asociados a cada intervalo.

Considérese el caso de emplear polinomios de grado 3 separados por un único nodo ξ. Se tiene entonces que f está definida por dos polinomios

$$f(X) = \begin{cases} \beta_{01} + \beta_{11}X + \beta_{21}X^2 + \beta_{31}X^3 & si \quad X < \xi \\ \\ \beta_{02} + \beta_{12}X + \beta_{22}X^2 + \beta_{32}X^3 & si \quad X \geq \xi \end{cases} \tag{3.7}$$

Ambos polinomios se pueden ajustar por mínimos cuadrados. Para el ajuste de la primera parte $\{\beta_{01}, \beta_{11}, \beta_{21}, \beta_{31}\}$ se usarán los datos de entrenamiento con $X < \xi$. Mientras que para la obtención de los coeficientes $\{\beta_{02}, \beta_{12}, \beta_{22}, \beta_{32}\}$, se tomarán las observaciones con $X \geq \xi$.

3.3. *Splines*

El principal inconveniente de los polinomios por partes es que pueden presentar discontinuidades o excesos de "flexibilidad" en los nodos (Figura 3.2). Esto genera que existan ciertas regiones poco fiables, concretamente los entornos de los nodos, debido a la gran variabilidad de la predicción: una observación a la derecha de un nodo puede dar lugar a una predicción muy dispar de otra observación relativamente cercana pero a la izquierda de dicho nodo.

Una forma de corregir esto es imponiendo la condición de continuidad. La manera más simple es forzar a cada polinomio a pasar por los puntos por los que pasan los contiguos a sus nodos (gráfica superior derecha Figura 3.2).

De esta manera se consigue la continuidad, pero f aún puede presentar grandes irregularidades cuando cambia de un intervalo a otro. Para conseguir la "suavidad" en los nodos se impone, al mismo tiempo, la continuidad de la derivada en los nodos. Estos son los *splines* (véase [6]). Un *M-spline* es un polinomio por partes de grado $M-1$ cuyas $M-2$ derivadas son continuas en cada nodo. Matemáticamente esto se indica como que un *M-spline* es de clase C^{M-2}.

Dado que el ojo humano no es capaz de apreciar la discontinuidad en los nodos a partir de la segunda derivada, los *splines* más comunes son los de orden $M = 4$, conocidos como *splines* cúbicos (Figura 3.2).

En general, una base para un *M-spline* con nodos $\{\xi_1, \ldots, \xi_K\}$ puede ser

$$\begin{aligned} &h_i(X) = X^{i-1}, i = 1, \ldots, M \\ &h_{M+l}(X) = (X - \xi_l)_+^{M-1}, l = 1, \ldots, K \end{aligned} \tag{3.8}$$

donde $(X - \xi)_+$ denota la parte positiva.

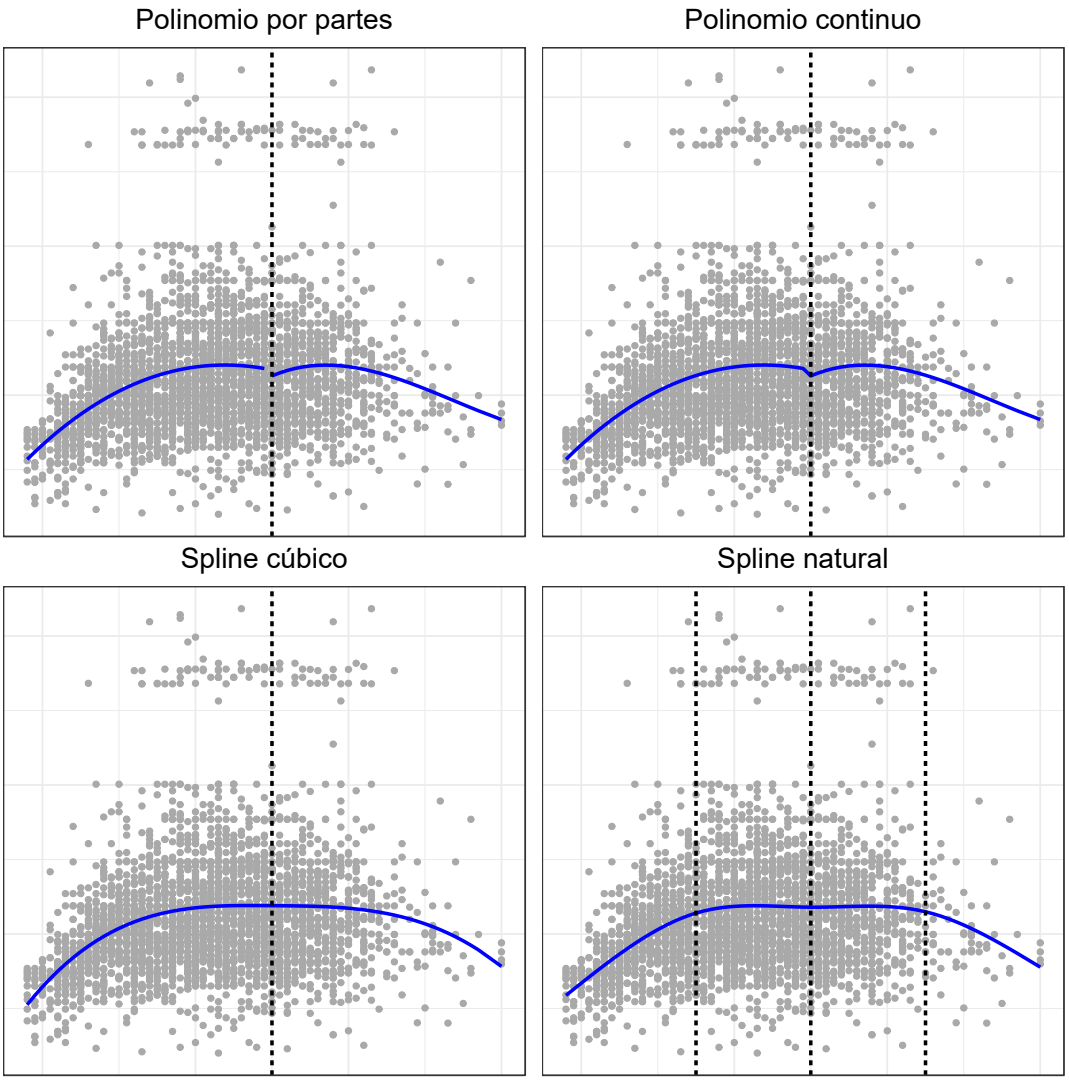

Figura 3.2: Ejemplo de modelos de regresión polinómica. Los puntos grises son los datos de entrenamiento a partir de los cuales se ajustaron los modelos. Las líneas discontinuas representan los nodos. Los polinomios son de grado 3. Para el spline natural (parte inferior derecha) se usaron dos nodos adicionales para suavizar los extremos.

Es decir, en un *spline* cúbico con K nodos el ajuste requiere estimar $K+4$ coeficientes. Como se explicó anteriormente, dado que el modelo es lineal para los predictores $\{h_1(X), \ldots, h_{K+4}(X)\}$, este ajuste puede realizarse a través de mínimos cuadrados.

El método de *splines* depende considerablemente del número y posición de nodos en los que se divide el dominio de la función. La elección de éstos no es trivial, pues como se comentó previamente, el número de nodos influye en los grados de libertad del ajuste y, en consecuencia, en la flexibilidad de la función. Una manera de realizar esta elección es tomar la cantidad de nodos para la cuál la suma de residuos al cuadrado del ajuste (RSS) sea mínima. Una vez hallado este número de grados de libertad, se calculan los cuantiles correspondientes para distribuir los nodos uniformemente.

Dado que en los extremos de los *splines* no se impone ninguna restricción para evitar la "flexibilidad", una opción usada frecuentemente para suavizar el *spline* en el primer y último intervalo es considerar rectas en estas partes, i.e, polinomios de grado 1. Esto se conoce como *spline natural* (veáse Figura 3.2).

3.4. *Splines* de suavizado

Además de la estrategia descrita previamente, existen otros procedimientos para definir *splines*. A continuación se expondrá el método de los *splines* de suavizado. Este planteamiento resulta de especial interés pues evita los problemas que pueden surgir en la elección de los nodos.

El problema consiste en encontrar una función f que minimice la suma de residuos al cuadrado $RSS(f) = \sum_{i=1}^{N} \{Y_i - f(X_i)\}^2$. Si no se impone ninguna restricción, el resultado sería una función que pase por todos los puntos $f(X_i) = Y_i \ \ \forall i = 1, \ldots, N$. Esta función tendría $RSS(f) = 0$ pero no sería en absoluto suave presentando demasiadas irregularidades.

Para conseguir un ajuste suave, en la RSS se incluye una una función de penalización

$$RSS(f, \lambda) = \sum_{i=1}^{N} \{Y_i - f(X_i)\}^2 + \lambda \int \left\{f''(t)\right\}^2 dt \tag{3.9}$$

La función de clase C^2 que minimice (3.9) se denomina *spline* de suavizado. Esta expresión está formada por dos términos.

- **Función de pérdida:** $\sum_{i=1}^{N} \{Y_i - f(X_i)\}^2$. Es una medida de la bondad del ajuste.
- **Función de penalización:** $\lambda \int \left\{f''(t)\right\}^2 dt$. Es la curvatura de f.

Como f' es una medida de la pendiente, f'' mide cuanto varía la pendiente. Por otro lado, $\int (f'')^2$ mide la variación de f' . De esta manera si f es suave, f' será aproximadamente constante y la integral tendrá un valor pequeño. De lo contrario, si f es muy flexible esta integral tenderá a tomar valores mayores.

El valor no negativo λ se denomina parámetro de penalización. Si λ tiene una valor cercano a 0 se penalizará poco la flexibilidad y el ajuste será poco suave. Si por el contrario, se dan valores grandes para este parámetro, la función del ajuste será más suave.

El resultado entonces es una función de clase C^2 definida por partes con tantos nodos como puntos, esto es, un *spline* cúbico natural con nodos $\{X1, \ldots, X_N\}$.

El *spline* de suavizado obtenido con este método no es el mismo que el *spline* cúbico natural obtenido por los métodos de regresión polinómica descritos. Como se explicó anteriormente, aumentar el número de nodos incrementa los grados de libertad del ajuste y, por tanto, la irregularidad de la función. Sin embargo, considerar un parámetro de penalización λ lleva a que los grados de libertad *efectivos* sean menores.

En el caso anterior de regresión polinómica, los grados de libertad son los parámetros libres, es decir, el número de coeficientes del ajuste, por lo que si que guardan relación con la flexibilidad de la función. Imponer una restricción, en este caso la función de penalización, lleva a que aunque el número de grados de libertad se mantenga, el modelo sea más suave. Luego el concepto de grados de libertad efectivos se puede entender como la influencia real de los grados de libertad sobre la suavidad de la función.

Por tanto, en este método ya no hay que realizar una elección de nodos. En cambio, se tiene que tomar un valor para λ en función de la penalización que se quiere dar a la flexibilidad del ajuste.

3.5. Caso práctico: cardiopatía

Todos estos modelos aditivos se pueden aplicar a problemas de clasificación modelando el logaritmo de las *odds* usando la función *logit*, tal y como se explicó en la regresión logística. De esta manera si se tienen p predictores para clasificar una observación, esto es, la variable de entrada es un vector $X = (x_1, \ldots, x_p)$, se usará el modelo

$$log\left(\frac{P(G=1|X)}{1-P(G=1|X)}\right) = \beta_0 + f_1(x_1) + \cdots + f_p(x_p) \tag{3.10}$$

donde f_i son funciones no necesariamente lineales obtenidas por los métodos de regresión polinómica, *splines*, *splines* de suavizado etc.

Así el modelo para el cálculo de las probabilidades a posteriori será tal que

$$P(G=1|X) = \frac{e^{\beta_0+f_1(x_1)+\cdots+f_p(x_p)}}{1+e^{\beta_0+f_1(x_1)+\cdots+f_p(x_p)}} \tag{3.11}$$

En el siguiente ejemplo se pretende comprobar la eficacia de realizar una prueba determinada para predecir si una persona es propensa a cierta enfermedad cardíaca. Para ello, se recogieron datos de personas sanas $G = 0$ y con la cardiopatía, $G = 1$. Concretamente, los datos tomados fueron la edad y el resultado del ritmo cardíaco máximo en dicha prueba. Las variable de entrada por tanto es un vector de dimensión 2 formado por los parámetros edad y ritmo cardíaco máximo.

En cuanto a la construcción y validación del modelo, los datos recogidos se dividieron en un conjunto de entrenamiento con el 80 % de las muestras (242 pacientes), para el ajuste, y otro conjunto de validación con el 20 % restante (61 muestras).

Tabla 3.1: Muestra aleatoria de cuatro datos del conjunto de entrenamiento. Un total de N=303 personas realizaron las pruebas, de las cuales 164 estaban sanas y 139 no. Se puede considerar una muestra equilibrada.

	Edad	RitmoMaximo	Diagnostico
#7	62	160	1
#157	51	173	1
#160	68	151	0
#235	54	163	0

La Figura 3.1 es el resultado de modelar la probabilidad de presentar la enfermedad en función de la edad. En este caso se usó una regresión polinómica de grado 3 con la única variable de entrada la edad de los pacientes. Si se realiza el mismo modelo para la variable del ritmo cardíaco máximo, se obtienen los resultados mostrados en el Tabla 3.2 y el modelo para la probabilidad a posteriori de la Figura 3.3.

Usando el método de *splines* cúbicos naturales incluyendo ambos predictores (Figura 3.4) en los datos del conjunto de validación se obtienen los datos mostrados en la matriz de confusión de la Figura 3.5 .

El modelo clasifica correctamente el $\frac{23+21}{61} = 72.13\,\%$ de las observaciones del conjunto de validación.

En la mayoría de casos, este modelo es considerablemente más preciso que el modelo logístico lineal descrito en el capítulo anterior dado que, precisamente, es una generalización de éste. En los GAM no se está limitado a considerar una transformación lineal, por lo que tienen una mayor capacidad de adaptación a la casuística del mundo real.

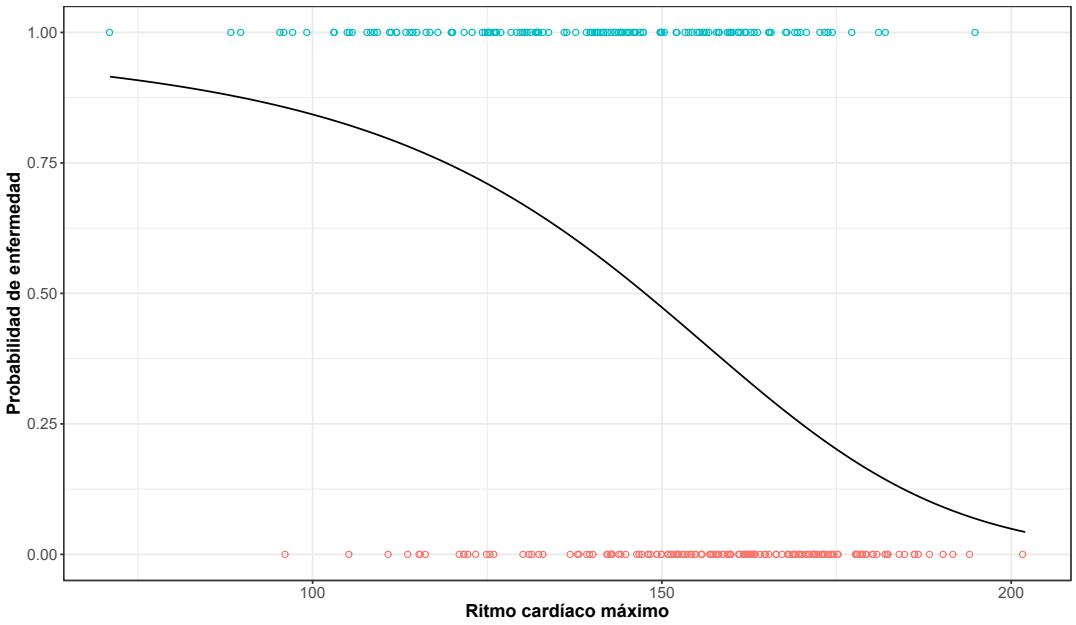

Figura 3.3: Regresión polinómica logística de grado 3 que modela la probabilidad de padecer una enfermedad solamente considerando el ritmo cardíaco máximo.

Como prueba de esto, el modelo lineal (tal y como se formuló en el Capítulo 2) clasifica este ejemplo con una precisión del 55.74 % como se muestra en la matriz de confusión de la Figura 3.5. En la predicción obtenida para el conjunto de validación se tiene un total de 23 falsos negativos, correspondiente a un 37.71 % de la muestra, en comparación con el 11.47 % obtenido con los *splines* cúbicos naturales.

Tabla 3.2: Estimación de los coeficientes, error estándar y *p-valor* de cada predictor para el modelo de regresión polinómica de grado 3. De la segunda fila en adelante se muestran los predictores correspondientes a las potencias de la variable ritmo máximo.

	Coeficientes	Error estándar	p-valor
(Intercept)	-0.2332787	0.1538029	0.1293329
poly(RitmoMaximo, 3)1	-16.9818607	3.1570245	0.0000001
poly(RitmoMaximo, 3)2	-3.3322205	3.2228206	0.3011617
poly(RitmoMaximo, 3)3	-3.0419886	4.0341776	0.4508167

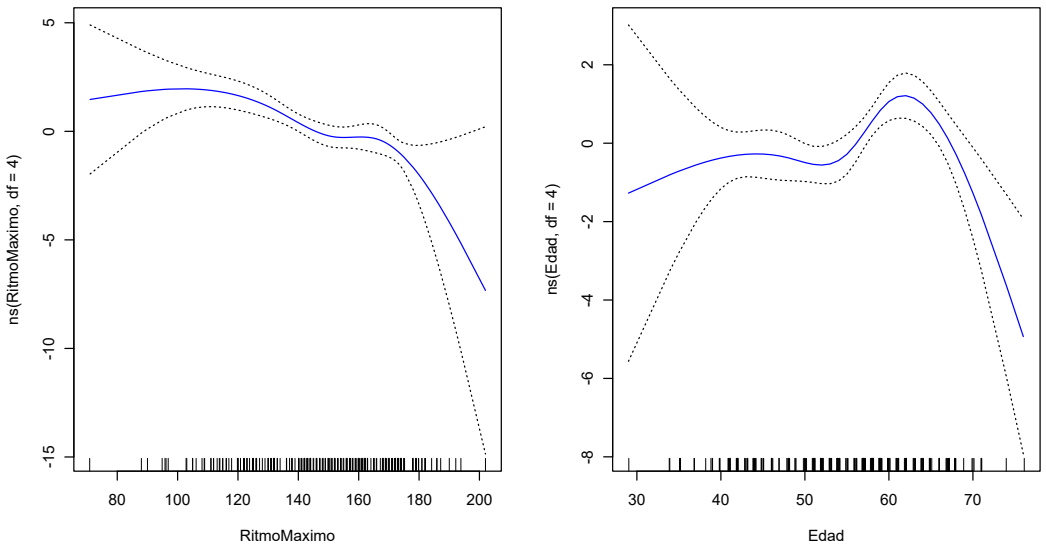

Figura 3.4: *Splines* cúbicos naturales para el modelo con ambos predictores.

		Diagnóstico Real	
		Sano	Enfermo
Predicción	Sano	23	7
	Enfermo	10	21

		Diagnóstico Real	
		Sano	Enfermo
Predicción	Sano	22	16
	Enfermo	11	12

Figura 3.5: Matrices de confusión para el conjunto de validación. Arriba se muestra la del del modelo de *splines* cúbicos naturales. Abajo la del modelo lineal.

Capítulo 4

K-vecinos más próximos

4.1. Introducción a los métodos *Kernel*

En este capítulo se expondrá el algoritmo de los K-vecinos más próximos o, por sus siglas en inglés, *kNN*. Este algoritmo pertenece a una clase de métodos, llamados *Kernel*, que se basan en hallar una función de clasificación ajustando distintos modelos para cada punto X_0 del dominio, es decir, para cada observación de la muestra de entrenamiento.

Teniendo en cuenta las dificultades con la flexibilidad surgidas en la regresión polinómica (véase el capítulo anterior), es razonable pensar que usar un modelo para cada punto llevaría a la misma problemática. Para evitar esto, en los siguientes métodos se usa una función de pesos $K_\lambda(X_0, X_i)$, denominada *Kernel*, que asigna un peso para cada X_i en función de su distancia a X_0 (véase [1, Capítulo 6]).

De esta manera, en el ajuste de los modelos para cada punto X_0 se tendrán más en cuenta las observaciones X_i cuanto más cercanas sean a X_0. En consecuencia, se logrará que el modelo final sea suave.

El parámetro λ del *Kernel* está asociado a la amplitud de los entornos a considerar. La elección de este parámetro se hará, principalmente, basándose en la muestra de entrenamiento.

4.2. El kNN

El algoritmo de los k-vecinos más cercanos es el método *Kernel* más elemental. Dada una observación X_0 de dimensión p que se pretende clasificar en una de las categorías $G = \{G_1, \ldots, G_L\}$, se calcula la distancia entre X_0 y el resto de observaciones de la muestra de entrenamiento X_i. Aunque se pueden considerar varias distancias en función del

problema a tratar, generalmente la más usada es la euclídea $d(X_0, X_i) = \sqrt{\sum_{j=1}^{p} (x_{0j} - x_{ij})^2}$.

A continuación, se seleccionan los k datos de entrenamiento más cercanos con sus respectivas clases. X_0 se clasificará en aquella clase G_i que sea la mayoritaria entre las k observaciones seleccionadas. En adelante, a este conjunto formado por los k datos más cercanos a X_0 también se le denominará *entorno* o *k-entorno* de X_0.

Una característica importante de este método que los distingue con respecto a los vistos hasta ahora es que se pueden usar para la clasificación no binaria sin recurrir a algoritmos externos (véase *Clasificación binaria*, Capítulo 1).

4.3. La elección de k

En este caso, el parámetro λ del que se habló anteriormente pasa a ser k, esto es, el número de observaciones entre las que se pretende comparar la observación a clasificar. Cuanto mayor sea k, mayores serán los entornos considerados.

En este método la elección del parámetro presenta algunas peculiaridades. Por ejemplo, para el caso binario resulta más adecuado tomar un valor impar para k con el fin de evitar posibles empates.

La opción más intuitiva es tomar $k = 1$, es decir, clasificar una observación en la clase correspondiente al dato de entrenamiento más cercano. El inconveniente de esta elección es que a medida que se reduce el valor de k, las predicciones se vuelven menos estables (algo parecido a los polinomios no suaves en la regresión polinómica) además de que se incrementa el riesgo de sobreajuste (i.e, que el modelo se adapte bien solamente al conjunto de entrenamiento y no para clasificar nuevas observaciones).

Por el contrario, para una k grande, el modelo será más estable a costa de incrementar el número de clasificaciones erróneas.

En general, para la elección de este parámetro, se aplicará el modelo al conjunto de validación (obtenido después de separar la muestra original, véase *Validación de los métodos* en Capítulo 1) varias veces con distintos valores de k. El valor para el cuál se cometan menos errores será el seleccionado.

4.4. La función de pesos

En cuanto a la función *Kernel* que asigna el peso ω_i al dato de entrenamiento X_i, se pueden hacer dos consideraciones.

Por un lado, no tener en cuenta las distancias dentro de los entornos. Es decir, una vez

construido el entorno de la observación con los k datos más cercanos, dar el mismo peso a todos los puntos en él.

Matemáticamente una posible formulación es considerar los conjuntos de índices I_X para los puntos de la muestra de entrenamiento no pertenecientes al k-entorno de X y J_X para los que sí. El *Kernel* asignaría los pesos $\omega_i = 0$ para $i \in I_X$ y $\omega_i = 1/k$ para $i \in J_X$.

El problema de este planteamiento es que, aún considerando un k impar, se podrían llegar a empates. Por ejemplo, si se tienen 3 clases y $k = 5$ se puede dar el caso de que dos puntos pertenezcan a una clase, dos a otra y uno a la última. Esta situación se podría resolver clasificando en una clase aleatoria entre las ganadoras por ejemplo, o más precisamente, incrementando el valor de k para el entorno donde se produce el empate (este procedimiento será el usado en el ejemplo práctico de este capítulo).

El otro posible enfoque es tener en cuenta las distancias también para los puntos dentro del entorno. La formulación de la función de pesos para este caso sería algo más compleja (véase [8]). De esta manera, la situación de empate descrita previamente se podría resolver asignando la categoría de los puntos a menor distancia, en promedio por ejemplo, de la observación.

4.5. Caso práctico: bodegas de vino

En el siguiente ejemplo se dispone de los resultados del análisis químico de la composición de un vino de una región de Italia producido por tres bodegas distintas etiquetadas como $G = \{1, 2, 3\}$.

En el análisis se midieron los valores de un total de 13 componentes para cada muestra como la graduación de alcohol, acidez, alcalinidad, magnesio... (véase Tabla 4.1)

Tabla 4.1: Muestra aleatoria de cuatro datos del conjunto de entrenamiento con los valores obtenidos de los cuatro primeros componentes. Se tiene un total de N=178 muestras de vinos de la cuales 59 han sido elaborados en la primera bodega, 71 en la segunda y 48 en la tercera.

	bodega	alcohol	acidez	alcanilidad	magnesio
#7	1	14.39	1.87	14.6	96
#47	1	14.38	3.59	16.0	102
#98	2	12.29	1.41	16.0	85
#155	3	12.58	1.29	20.0	103

Para este ejemplo se dividió el conjunto de muestra original en dos. Una parte, formada por el 20 % de las muestras (36 observaciones concretamente), es el conjunto de validación que se usará en el modelo kNN para comparar las clases predichas con las reales y hallar el error. La otra parte, formada por el 80 % de las muestras (142 datos), forma conjunto de entrenamiento cuyas categorías son las comparadas en los k-entornos de las observaciones del conjunto de validación para asignar las predicciones.

El algoritmo usado en este modelo kNN resuelve los empates ajustando el tamaño de cada entorno en caso de que se produzca uno. Como se explicó previamente, fijado un k, una observación se clasificará en aquella clase que sea la mayoritaria entre los datos de entrenamiento en su k-entorno. En caso de empate, se incrementa en 1 el valor de k, es decir, se realiza el mismo procedimiento pero incluyendo el siguiente dato de entrenamiento más cercano a la observación. Si el nuevo vecino desempata concluye el proceso. En caso contrario, se vuelve a incrementar k en 1 hasta que no se de un empate.

En la Figura 4.1 se muestra la proporción de observaciones bien clasificadas por este algoritmo para distintos valores de k.

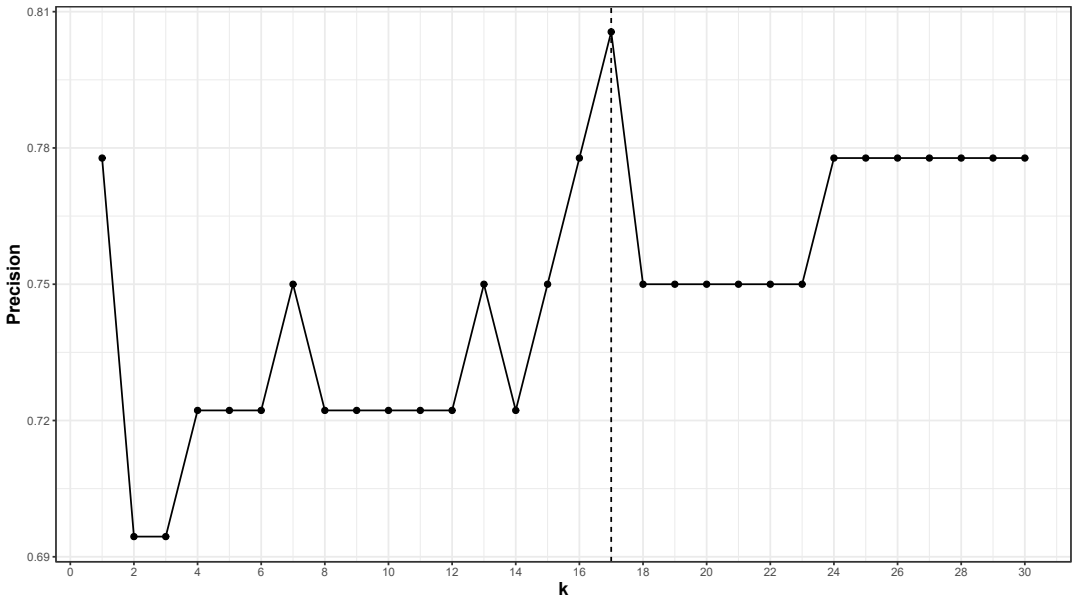

Figura 4.1: Precisión del modelo kNN en la clasificación del conjunto de validación para distintos valores de k. El modelo con menor error se tiene para k=17 (línea discontinua).

El modelo con $k = 17$ es el que mejor clasifica el conjunto de validación con una precisión del 80.56 %. Los resultados se muestran en la matriz de confusión (Figura 4.2).

		Bodega Real		
		1	2	3
Predicción	1	10	0	0
	2	0	14	5
	3	2	0	5

Figura 4.2: Matriz de confusión del conjunto de validación para el modelo kNN con k=17.

Solamente 2 muestras de la bodega $G = 1$ y 5 de la $G = 3$ fueron mal clasificadas.

4.6. Caso práctico: cardiopatía con kNN

Con intención de comparar los modelos de clasificación expuestos en la memoria, se aplicará este algoritmo kNN al ejemplo del capítulo anterior (véase *Caso práctico: cardiopatía*, Capítulo 3).

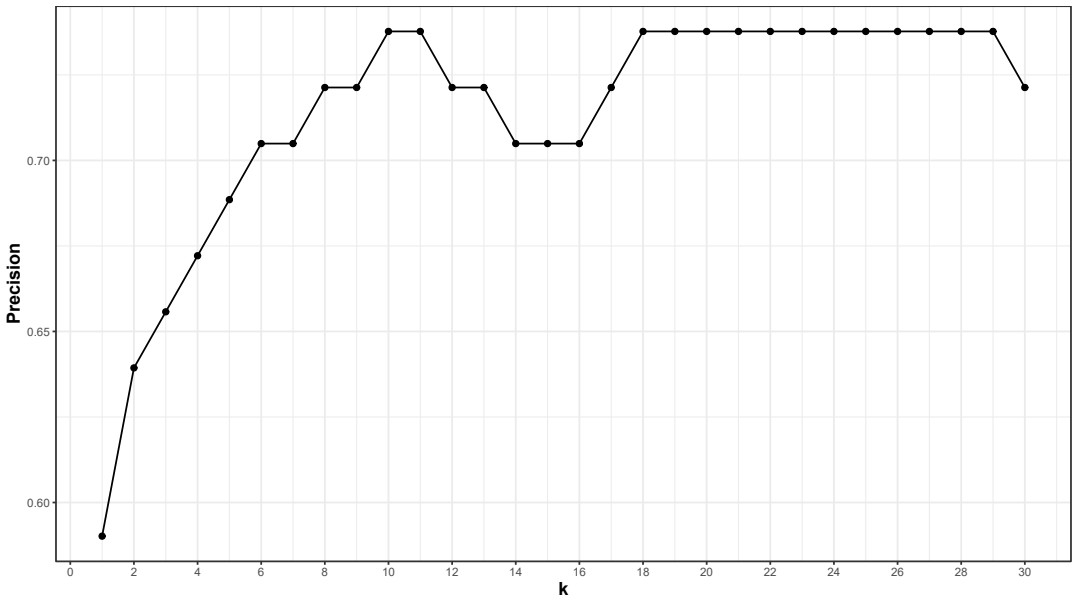

Figura 4.3: Precisión del modelo kNN en la clasificación del conjunto de validación para distintos valores de k. En este caso, el error se minimiza para más de una valor de k.

En este caso, el valor de k para el cual se minimiza el error de clasificación en el conjunto de validación no es único. No obstante, se tomará uno de ellos (concretamente $k = 10$) que minimice los falsos negativos obtenidos, dada la naturaleza del problema a tratar.

Los resultados obtenidos (Figura 4.4) tienen una precisión muy parecida a la del modelo GAM, concretamente 73.77 %.

		Diagnóstico Real	
		Maligno	Benigno
Predicción	Maligno	27	10
	Benigno	6	18

Figura 4.4: Matriz de confusión para el conjunto de validación con $k = 10$.

Aun así, para este caso sigue siendo preferible el clasificador modelado con *splines* puesto da lugar a menos falsos negativos. Precisamente, este es el error que se pretende minimizar en un problema del tipo de diagnóstico médico. Resulta mucho más crítico considerar sana a una persona enferma que el caso contrario.

Capítulo 5

Métodos *Kernel* y SVM

Las máquinas de vector soporte, o SVM por sus siglas en inglés, son una clase de algoritmos desarrollados por Vladimir Vapnik y su equipo en los 90. Originalmente este modelo estaba pensado para la clasificación binaria aunque, gracias a su gran fiabilidad, pronto se adaptó para problemas de clasificación en más categorías e, incluso, regresión.

En líneas generales, el SVM se basa usar el conjunto de entrenamiento para dividir el espacio de la variable de entrada en regiones asociadas a cada clase. Después, una nueva observación se clasifica en función de la región a la que pertenezca. Separar el espacio de entrada para definir estas regiones es, precisamente, el problema central de este método.

Para llegar a comprender el algoritmo completo, primero se expondrán una serie de conceptos básicos en los que se fundamentan los SVM.

5.1. Hiperplanos de separación

Supóngase un problema de clasificación binaria y un conjunto de entrenamiento $\{(X_i, G_i)\}_{i=1}^N$ con $X_i \in \mathbb{R}^p$ y $G_i \in \{-1, 1\}$. La estrategia de los SVM para clasificar es dividir el espacio p-dimensional de entrada en dos regiones. Se pretende que una de ellas contenga las observaciones de entrenamiento X_i con $G_i = -1$, mientras que los datos con $G_i = 1$ pertenezcan a la otra región. Formalmente, esto se consigue definiendo un hiperplano que separe el espacio de entrada en dos semiespacios.

En un espacio de dimensión p un hiperplano es un espacio afín de dimensión $p - 1$

$$\left\{X : \beta_0 + X^T\beta = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = 0\right\} \tag{5.1}$$

donde se tomará $\|\beta\| = 1$. (véase [1, Capítulo 4])

Un hiperplano divide el espacio de entrada en dos semiespacios. Definiendo la función $f(X) = \beta_0 + X^T\beta$ asociada a dicho hiperplano, estos semiespacios son $\{X : f(X) > 0\}$ y

$\{X : f(X) < 0\}$. La regla de clasificación que determina f o, equivalentemente, el hiperplano elegido será

$$G(X) = sgn(f(X)) \tag{5.2}$$

donde sgn denota la función signo con el convenio $sgn(0) = 1$.

Tal y como se definió f, el valor de $f(X)$ es la distancia de la observación X al hiperplano $f(X) = 0$ (véase [1, Capítulo 4]), por lo que además, da una idea de la fiabilidad de la predicción puesto que cuanto más se aleje la observación del hiperplano, más certeza se tiene de que la predicción sea correcta.

La elección de los parámetros β_0 y β se hará en función del conjunto de entrenamiento, de tal forma que el hiperplano definido sea capaz de separar, en la medida de lo posible, los datos de cada clase.

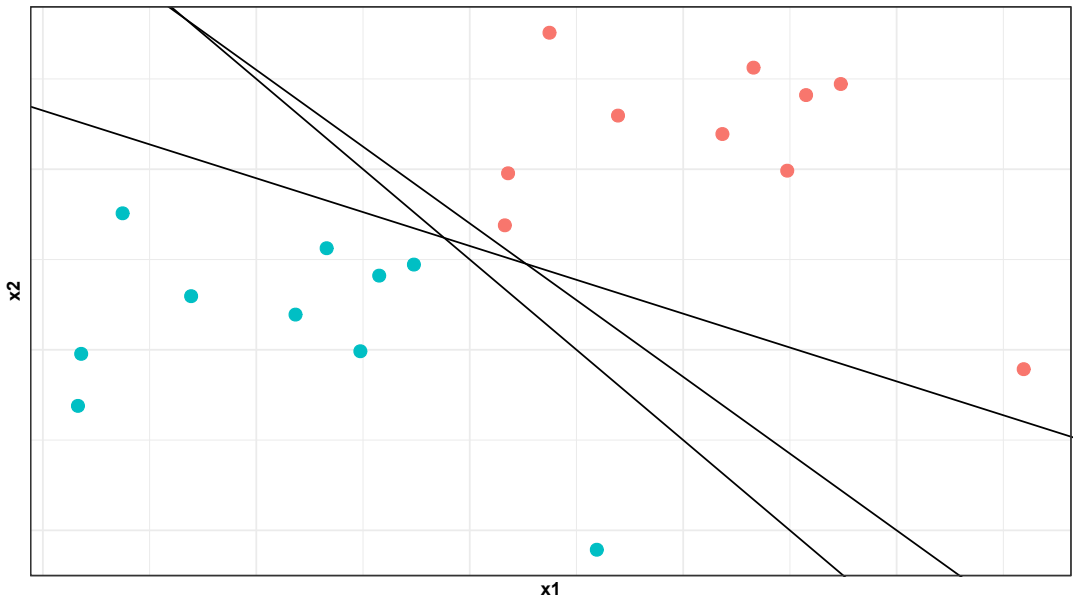

Figura 5.1: Ejemplo de tres hiperplanos de separación en un espacio de dimensión 2. Las categorías de los datos de entrenamiento están representadas por distintos colores. La función de predicción inducida por cada hiperplano clasificará las nuevas observaciones que estén por encima de la respectiva recta como "Naranja" y como "Azul" las que estén por debajo. Para facilitar la visualización, en adelante, los ejemplos gráficos serán con datos bidimensionales, ya que así, los hiperplanos son rectas.

5.1.1. Clases separables

El caso más sencillo es que las clases sean linealmente separables, es decir, que exista un hiperplano que separe perfectamente los datos de ambas clases (véase Figura 5.1). Esta condición equivale matemáticamente a que exista una f tal que $f(X_i)G_i > 0 \ \ \forall i$.

Para este caso existen infinitas soluciones en el sentido que es posible hallar infinitos hiperplanos que separen perfectamente las clases (Figura 5.1). Existen distintos algoritmos para obtener hiperplanos de separación como por ejemplo el *perceptrón* (véase [9]).

El planteamiento por tanto consistiría en calcular cuál de todos los hiperplanos de separación posibles es el óptimo, esto es, aquél que además de separar las dos clases, maximice la distancia a los puntos más próximos de cada una. La distancia de un hiperplano al dato más cercano se conoce como *margen* y se denotará como M. El hiperplano óptimo es el que tiene un mayor margen. Con esta elección, además de obtener una solución única, se consigue reducir el error de clasificación lo máximo posible.

Como se tiene que f da la distancia de un punto al hiperplano $f(X) = 0$, si se considera X_0 uno de los puntos más cercanos al hiperplano, entonces $M = f(X_0)$. En definitiva, la formulación matemática es un problema de optimización

$$\begin{aligned} \max_{\beta,\beta_0,\|\beta\|=1} \quad & M \\ \text{s.a.} \quad & G_i\left(X_i^T\beta + \beta_0\right) \geq M \ , \ i = 1, \ldots, N \end{aligned} \tag{5.3}$$

El cuál es posible resolver usando métodos de optimización numérica (como se explica en [9]).

5.1.2. Clasificador SV

La situación descrita previamente, a pesar de ser ideal, no es la más realista. Tener una muestra de datos que se puedan separar perfectamente con un hiperplano no es lo habitual en absoluto. Además, incluso dándose esta condición, el método presenta algunos inconvenientes.

Por una parte, una variación mínima en el conjunto de entrenamiento puede variar en gran medida el hiperplano a considerar. Además el hecho de que el hiperplano de separación óptimo se ajuste perfectamente a la muestra, tiende a ocasionar problemas de sobreajuste.

Por estos motivos es interesante formular una generalización del método anterior, llamada clasificador SV, usando lo que se denomina *margen blando*.

Tal y como se explica en [1, Capítulo 4] el problema (5.3) es equivalente al siguiente

$$\begin{aligned} \min_{\beta,\beta_0} \quad & \|\beta\| \\ \text{s.a.} \quad & G_i \left(X_i^T \beta + \beta_0\right) \geq 1 \ , \ i = 1, \ldots, N \end{aligned} \tag{5.4}$$

con $M = 1/\|\beta\|$

La idea es seguir maximizando el margen M del hiperplano pero permitiendo a algunos puntos que estén en el margen incorrecto para su clase como se muestra en la Figura 5.2.

Una forma de solucionar esto es introduciendo variables de holgura ξ_i. La variable ξ_i se define de tal forma que sea la cantidad proporcional por la cuál la predicción $f(X_i)$ está en lado incorrecto de su margen.

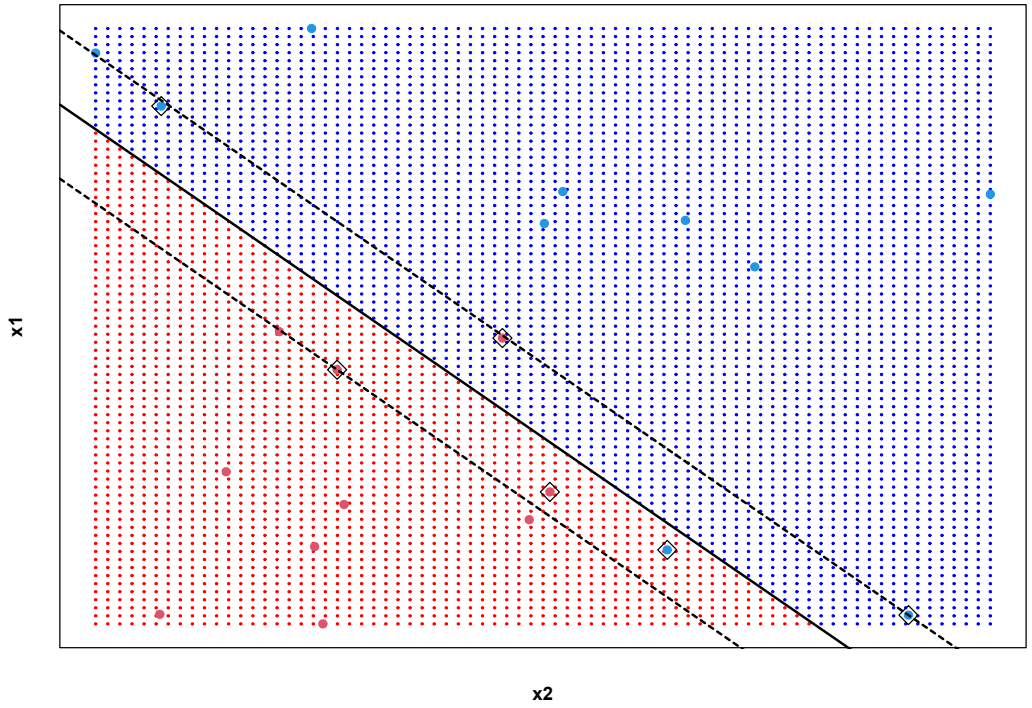

Figura 5.2: Ejemplo de hiperplano separador con margen blando para un caso de clases no separables. En este caso se permite que algunos puntos estén en el lado incorrecto del margen. El margen es la distancia del hiperplano a las lineas discontinuas. Los vectores soporte son los indicados con un cuadrado.

Por tanto, al problema de optimización (5.4) hay que añadir una nueva restricción para incluir cuanto se permite fallar en la clasificación de cada observación. Como se explica

en [1, Capítulo 11] esta restricción equivale a introducir una constante C y, finalmente, formular el problema convexo de optimización

$$\begin{aligned} \min_{\beta,\beta_0} \quad & \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.a.} \quad & \xi_i \geq 0 \ , \ G_i\left(X_i^T\beta + \beta_0\right) \geq 1 - \xi_i \ , \ i = 1, \ldots, N \end{aligned} \tag{5.5}$$

donde C es el parámetro de costo. Cuanto más cercano es C a 0, menos se penalizan las clasificaciones erróneas y más datos pueden estar del lado incorrecto del margen. El caso de las clases separables corresponde a $C = \infty$.

En conclusión, la solución a este problema es un hiperplano $\hat{f}(X) = X^t\hat{\beta} + \hat{\beta}_0$ que separa de forma óptima los datos considerando un costo C para determinar en que medida se permitirá que ciertas observaciones estén del lado incorrecto del hiperplano. La función de clasificación será por tanto $\hat{G}(X) = sgn(\hat{f}(X))$.

Los datos que se encuentran en el propio margen, tanto en el lado correcto como no, son lo que se denominan *vectores soporte* (véase Figura 5.2). Estos datos son los que determinan, en definitiva, el hiperplano elegido y, por tanto, la función de clasificación.

La elección de C regula el modelo en el sentido que a medida que se aproxima a 0, el margen es cada vez mayor incrementándose el número de vectores soporte. De esta manera, se incrementa el sesgo pero se reduce la varianza. Contrariamente, para valores grandes de C el margen disminuye, teniéndose menos vectores soporte por lo que se reduce el sesgo pero incrementa la varianza.

Para el caso no binario, cuando se tienen L clases $G = \{1, \ldots, L\}$, a cada clase se le asocia un hiperplano $f_i(X) = 0$. Después, una nueva observación se clasifica en la categoría cuyo hiperplano este más alejado, obteniendo como función de clasificación

$$G(X) = \arg\max_{1\leq i\leq L} f_i(X) \tag{5.6}$$

Este escenario se explica más detalladamente en [12].

5.2. SVM

El método del clasificador SV resulta altamente eficiente cuando la separación de los datos de cada clase es posible a través de un hiperplano, incluso cuando la frontera entre las regiones es algo más "difusa" y se mezclan observaciones de varias clases. El problema surge cuando los datos de una muestra siguen un patrón por el que no es posible en absoluto separarlos linealmente por hiperplanos.

Como se hizo en el capítulo 3, un planteamiento para flexibilizar las fronteras es considerar una base de funciones $h_m(X)$, $m = 1, \ldots, M$. De esta forma, se ajustaría el clasificador SV con las variables de entrada $h(X_i) = (h_1(X_i), \ldots, h_m(X_i))$, $i = 1, \ldots, N$ y se obtendría una función de separación (ya no necesariamente lineal) $\hat{f}(X) = h(X)^T\hat{\beta} + \hat{\beta}_0$ con el mismo clasificador que antes (5.2).

El SVM es una generalización de esta idea usando funciones *Kernel*. Estas funciones permiten considerar el conjunto de datos en un espacio de dimensión mayor al original. De esta manera, conjuntos de observaciones que no son separables linealmente pueden si serlo en un espacio con más dimensiones.

De este modo se define un hiperplano de separación en este nuevo espacio y, a continuación, se proyecta en el original. Las fronteras lineales del espacio aumentado se transforman en no lineales cuando se proyectan en el espacio inicial.

Los problemas de optimización descritos para el cálculo de los hiperplanos se resuelven a través del método de los multiplicadores de Langrange ya que facilita enormemente su implementación en ordenadores. En síntesis, este procedimiento reduce un problema con restricciones a uno sin ellas pero con más variables. Para ello, se define una nueva función L, llamada *Lagrangiana*, cuya optimización resulta más sencilla (véase [13, Capítulo 7]).

Como se explica en [1, Capítulo 11], si al problema (5.5) se aplican las transformaciones de una base h y usando la notación del producto escalar $X^TX = \langle X, X\rangle$, se llega a que la solución por el método de los multiplicadores de Lagrange es la función

$$f(X) = h(X)^T\beta + \beta_0 = \sum_{i=1}^{N} \alpha_i G_i \langle h(X), h(X_i)\rangle + \beta_0 \tag{5.7}$$

donde los coeficientes α_i son los multiplicadores de Lagrange.

Tanto en el Lagrangiano como en la solución (5.7) la variable de entrada solamente aparece a través de $\langle h(X), h(X_i)\rangle$, esto es, el producto escalar de la nueva observación y los puntos del conjunto de entrenamiento. Es precisamente este producto escalar el que se reemplazará por una función *Kernel*.

5.2.1. El truco *Kernel*

Dado un espacio de entrada X se puede considerar una transformación $h : X \to \nu$ que lleva los datos de entrada en X a un espacio ν de mayor dimensión. Tomando la transformación adecuada, en este nuevo espacio los datos $h(X)$ pueden ser separados por hiperplanos. A primera vista, podría parecer que tal estrategia, aunque eficaz, complicaría su implementación en un algoritmo de clasificación.

Como se observa en (5.7), para obtener la función f de separación, es necesario considerar el producto escalar de los datos, evaluándolos previamente con una transformación determinada.

En este contexto se define una función *Kernel* $k : X \times X \to \mathbb{R}$ como una función simétrica y semi-definida positiva que evalúa la relación de pares de puntos del espacio X equivalentemente al producto escalar de las imágenes de estos puntos por cierta transformación $h : X \to \nu$

$$k(X, X') = \langle h(X), h(X') \rangle \tag{5.8}$$

Esto es, una función *Kernel* devuelve el producto escalar de dos elementos en X a los que se le aplicó previamente una transformación h que lleva dichos elementos de X a un espacio ν de dimensión mayor.

Considerando esto en la solución (5.7) se tiene

$$f(X) = \sum_{i=1}^{N} \alpha_i G_i k(X, X_i) + \beta_0 \tag{5.9}$$

Esencialmente, el planteamiento consiste en definir el producto escalar bajo cierta transformación, que resulta adecuada para la separación de los datos, en vez de aplicar la transformación y después calcular el producto escalar.

La clave es que para hallar la función de decisión (5.9) no se necesita conocer completamente la transformación $h(X)$, o dicho de otro modo, no es necesario llevar todos los datos al espacio de dimensión mayor, lo que sería muy costoso computacionalmente. En su lugar, para clasificar una observación solamente se requiere conocer su producto escalar con los datos de entrenamiento, esto es, N evaluaciones de la función *Kernel*, lo que resulta mucho más eficiente.

5.2.2. Funciones *Kernel*

Existen infinidad de funciones *Kernel*, algunas de las más usadas en el SVM (véase [1, Capítulo 11]) son:

- ***Kernel* polinómico de grado d:**

$$k(X, X') = \big(c + \langle X, X' \rangle\big)^d \tag{5.10}$$

  Cuando $d = 1$ y $c = 0$ este *Kernel* es lineal y equivale al producto escalar usual empleado en el clasificador SV. Para $d > 1$, las funciones de separación obtenidas son polinomios de grado d (por tanto no lineales). Este tipo de funciones *Kernel* son populares en problemas de procesamiento de imágenes por ejemplo.

- ***Kernel* gaussiano:**

$$k(X, X') = e^{-\gamma \|X - X'\|^2} \tag{5.11}$$

Es un *Kernel* de carácter general, frecuentemente usado cuando no se dispone de ningún conocimiento previo acerca de los datos. El parámetro γ regula la flexibilidad de la función de separación, cuanto más pequeño, más lineales son las fronteras. La gran ventaja de este tipo de *Kernel* es su versatilidad, puesto que ajustando γ se pueden lograr tanto funciones de separación lineales simples como fronteras mucho mas irregulares y complejas.

5.3. Caso práctico: cáncer de mama

Para el siguiente ejemplo se disponen de unos datos de entrenamiento obtenidos de un grupo de 569 mujeres que presentan un tumor en el pecho, algunos malignos (etiquetados como $G = 0$) y otros benignos ($G = 1$). Para cada paciente se midieron un conjunto de 30 atributos relacionados con la morfología y constitución del tumor: radio, textura, perímetro, área... (véase Tabla 5.1)

El objetivo es construir un modelo que pueda diagnosticar si el cáncer de mama de un individuo es maligno en base al valor medido de los 30 atributos de su tumor.

Tabla 5.1: Muestra aleatoria de cuatro datos del conjunto de entrenamiento con los cuatro primeros atributos de los 30 usados para ajustar el modelo. En este ejemplo, se tiene un total de N=569 mujeres diagnosticadas con 212 casos de tumores malignos y 357 benignos.

	diagnostico	radio	textura	perimetro	area
#7	0	18.250	19.98	119.60	1040.0
#35	0	16.130	17.88	107.00	807.2
#235	1	9.567	15.91	60.21	279.6
#457	1	11.630	29.29	74.87	415.1

En este ejemplo se usará el *Kernel* gaussiano (5.11). Para ello, es necesario fijar dos parámetros: el costo C, para penalizar las observaciones en el lado incorrecto de la frontera de decisión, y γ, que regula la flexibilidad de dicha frontera.

Con respecto a la validación del modelo, se selecciona aleatoriamente un 80 % de las muestras (456 exactamente) para formar el conjunto de entrenamiento para ajustar el

modelo. El 20 % restante (113 datos) forma el conjunto *test* que se usa para validar la función de clasificación obtenida.

Para realizar la elección de C y γ, se toman distintos valores y se compara el error cometido por el modelos en la clasificación del conjunto de validación. Como se observa en la Figura 5.3 se tiene que, para este caso, el valor óptimo de los parámetros es $C = 1$ y $\gamma = 0.5$.

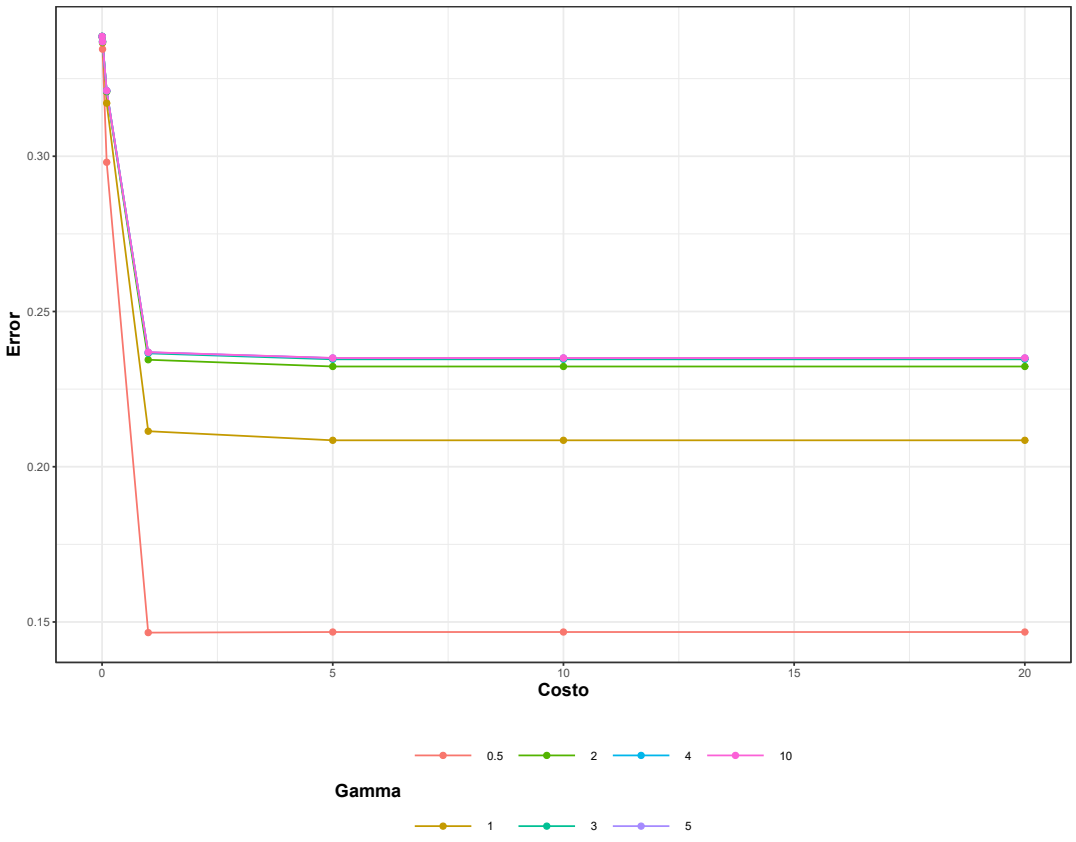

Figura 5.3: Comparación del error cometido en la clasificación del conjunto de validación tomando distintos valores para los parámetros C y γ.

Finalmente el modelo obtenido tiene una precisión de 87.61 %. La matriz de confusión del conjunto test en la partición original (i.e, con el conjunto de entrenamiento que ajustó el modelo) es la mostrada en la Figura 5.4.

		Diagnóstico Real	
		Maligno	Benigno
Predicción	Maligno	34	10
	Benigno	4	65

Figura 5.4: Matriz de confusión para el conjunto de validación.

5.4. Caso práctico: cardiopatía con SVM

Para el ejemplo de la cardiopatía del Capítulo 3 se ajustarán tanto el SVM con *kernel* gaussiano como el lineal.

Para el primer caso, se obtiene que los parámetro óptimos son $C = 5$ y $\gamma = 2$ (Figura 5.5).

En la clasificación del conjunto de validación se obtienen los resultados mostrados en las matrices de confusión de la Figura 5.6. Concretamente para el modelo lineal se obtiene una precisión del 63.93 % mientras que para el modelo con *gaussiano* se clasifican bien solamente el 57.38 % de las muestras del conjunto de validación.

Esto no significa que el SVM o el kNN sean métodos de clasificación estrictamente peores que el modelo logístico con bases de *splines*. El caso práctico de la cardiopatía es un ejemplo donde se tiene un número relativamente bajo de muestras de entrenamiento (242 después de la partición para la validación cruzada) con pocos predictores y además los datos presentan un cierto carácter lineal (los residuos son bajos al ajustar un modelo lineal). Este es el contexto perfecto donde los modelos logísticos logran un buen desempeño.

Mientras, los algoritmos como el kNN o el SVM son más adecuados para conjuntos grandes de datos, de alta dimensión (como los casos prácticos de los capítulos 4 y 5) donde las observaciones se alejan de un posible modelo más definido como el lineal.

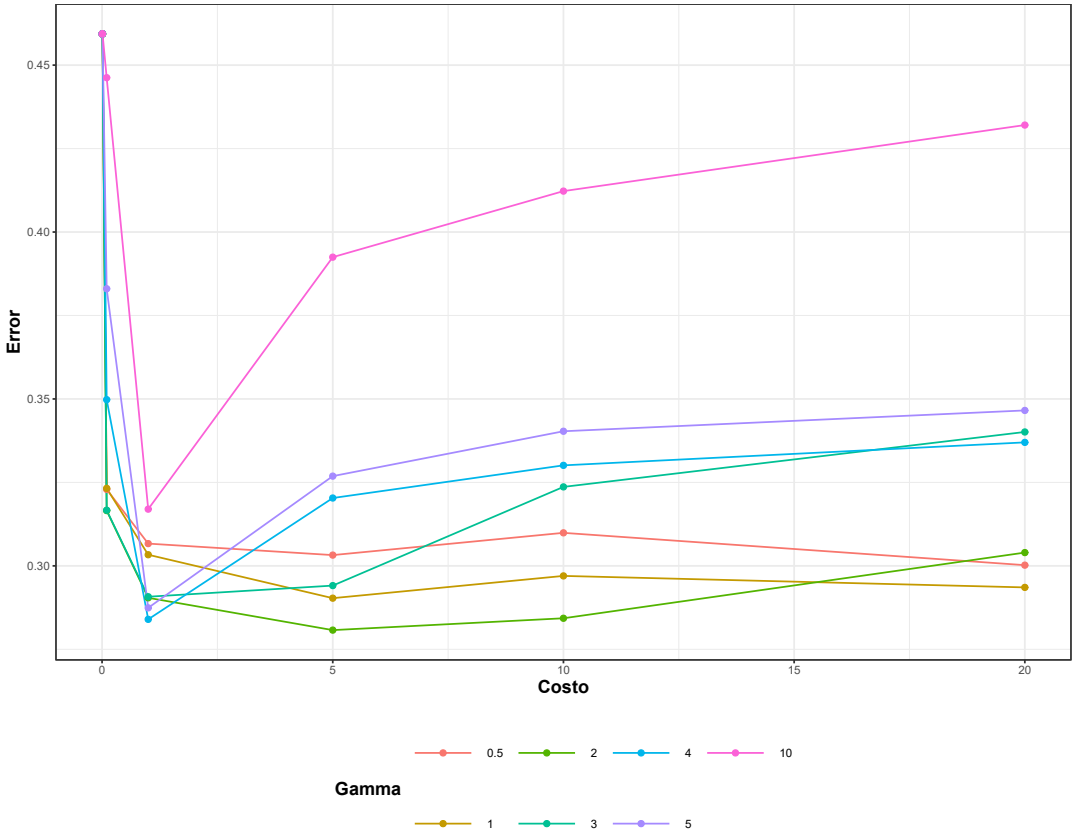

Figura 5.5: Comparación del error cometido en la clasificación del conjunto de validación en el ejemplo de la cardiopatía tomando distintos valores para los parámetros C y γ.

		Diagnóstico Real	
		Sano	Enfermo
Predicción	Sano	21	12
	Enfermo	14	14

		Diagnóstico Real	
		Sano	Enfermo
Predicción	Sano	24	9
	Enfermo	13	15

Figura 5.6: Matrices de confusión para el conjunto de validación. Arriba se muestra la del del modelo SVM lineal. Abajo la del modelo de *Kernel* gaussiano con $C = 5$ y $\gamma = 2$.

5.5. Caso práctico: sonar

En este ejemplo se tiene un conjunto de datos obtenidos de las ondas rebotadas que produce un sonar en una roca ($G = 1$) o un cilindro metálico que simula una mina ($G = 0$). El objetivo es construir un modelo para un sonar naval que sea capaz de detectar minas diferenciándolas de las rocas marinas.

Los datos recogidos por el sonar para cada muestra son un conjunto de frecuencias obtenidas del rebote de una señal original emitida en diversas condiciones y ángulos. En concreto, para cada muestra se emitió una onda en 60 condiciones distintas (etiquetadas como $\{V1, V2, \ldots, V60\}$). Por tanto, en este problema, la variable de entrada es un vector de dimensión $p = 60$. En la Tabla 5.2 se muestran algunas observaciones con los cuatro primeros predictores del total de 60 con los que se ajustaron los modelos.

Tabla 5.2: Muestra aleatoria de cuatro datos con los cuatro primeros atributos. En este ejemplo, se tiene un total de N=208 objetos clasificados de los cuales 111 son minas y 97 son rocas.

	V1	V2	V3	V4	clase
#5	0.0762	0.0666	0.0481	0.0394	1
#12	0.0123	0.0309	0.0169	0.0313	1
#105	0.0307	0.0523	0.0653	0.0521	0
#176	0.0294	0.0123	0.0117	0.0113	0

Como en el resto de ejemplos, la muestra de datos original se divide en un conjunto de validación del 20 %, con 41 datos, y uno de entrenamiento del 80 % con las 167 observaciones restantes.

Para este ejemplo se usaron varios modelos (Tabla 5.7). Por un lado se ajustaron dos modelos logísticos, uno lineal y otro con *splines* cúbicos naturales. Por otro lado, se modeló un SVM con *kernel* gaussiano.

Este es un ejemplo donde claramente los datos no presentan un carácter lineal, pues como se ve en los resultados (Tabla 5.7) los modelos logísticos obtiene una precisión baja: 65.85 % el lineal y 68.29 % el GAM, no mejorando apenas el modelo. En cambio, el SVM mejora significativamente esos resultados clasificando correctamente el 87.81 % de las observaciones.

		Clase Real: Mina	Clase Real: Roca
Predicción	Mina	16	8
	Roca	6	11

		Clase Real: Mina	Clase Real: Roca
Predicción	Mina	16	7
	Roca	6	11

		Clase Real: Mina	Clase Real: Roca
Predicción	Mina	20	2
	Roca	3	16

Figura 5.7: Matrices de confusión para el conjunto de validación. Arriba se muestran las de los modelos logístico: a la izquierda el lineal y la derecha el GAM. Abajo la del modelo SVM con *Kernel* gaussiano.

Bibliografía

[1] Hastie, T., Tibshirani, R., Friedman, J. H. (2008). *The elements of statistical learning: data mining, inference, and predicition.* 2nd ed. New York: Springer.

[2] James, G., Witten, D., Hastie, T.,Tibshirani, R. (2013). *An introduction to statistical learning : with applications in R.* New York: Springer.

[3] Faraway, J. J. (2016). *Linear Models with R.* Amsterdam University Press.

[4] Diez, D., Çetinkaya-Rundel, M., Barr, C. (2019). *OpenIntro Statistics: Fourth Edition.* OpenIntro.

[5] Lever, J., Krzywinski, M., Altman, N. (2016). *Points of Significance: Logistic regression.* Nature Methods.

[6] Durbán, M. (2013). *Modelos Aditivos Generalizados con P-splines.* Universidad Carlos III de Madrid.

[7] Hastie, T., Tibshirani, R. (1990). *Generalized Additive Models.* Chapman and Hall/CRC.

[8] Richard J. Samworth. (2012) *Optimal weighted nearest neighbour classifiers..* Annals of Statistics. 40 (5) 2733 - 2763.

[9] Kowalczyk, A. (2017). *Support Vector Machines Succinctly.* Syncfusion.

[10] Carmona, E. J. (2016). *Tutorial sobre Máquinas de Vectores Soporte (SVM).*

[11] Statnikov, A. R., Aliferis, C. F., Hardin, D. P., Guyon, I. (2013). *A Gentle Introduction to Support Vector Machines in Biomedicine: Case studies and benchmarks.* World Scientific.

[12] Angulo, C., Parra, X., Catala, A. (2003). *A support vector machine for multiclass classification.*

[13] Luenberger, D.G. (1969). *Optimization by Vector Space Methods.* New York: John Wiley and Sons