Notas sobre Regresión Logística
Breve introducción con aplicaciones

Carlos E Martinez-Rodriguez

Contents

Chapter 1

Principios

1.1 Análisis de Regresion Lineal (RL)

En muchos problemas hay dos o más variables relacionadas, para medir el grado de relación se utiliza el **análisis de regresión**. Supongamos que se tiene una única variable dependiente, y, y varias variables independientes, $x_1, x_2, \ldots, x_n$. La variable y es una varaible aleatoria, y las variables independientes pueden ser distribuidas independiente o conjuntamente. A la relación entre estas variables se le denomina modelo regresión de y en $x_1, x_2, \ldots, x_n$, por ejemplo $y = \phi(x_1, x_2, \ldots, x_n)$, lo que se busca es una función que mejor aproxime a $\phi(\cdot)$.

Supongamos que de momento solamente se tienen una variable independiente x, para la variable de respuesta y. Y supongamos que la relación que hay entre x y y es una línea recta, y que para cada observación de x, y es una variable aleatoria.

El valor esperado de y para cada valor de x es

$$E(y|x) = \beta_0 + \beta_1 x, \tag{1.1}$$

β_0 es la ordenada al orígen y β_1 la pendiente de la recta en cuestión, ambas constantes desconocidas.

Supongamos que cada observación y se puede describir por el modelo

$$y = \beta_0 + \beta_1 x + \epsilon \tag{1.2}$$

donde ϵ es un error aleatorio con media cero y varianza σ^2. Para cada valor y_i se tiene ϵ_i variables aleatorias no correlacionadas, cuando se incluyen en el modelo 1.2, este se le llama *modelo de regresión lineal simple*. Supongamos además que se tienen n pares de observaciones $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, estos datos pueden utilizarse para estimar los valores de β_0 y β_1. Esta estimación es por el **métodos de mínimos cuadrados**.

Entonces la ecuación (1.2) se puede reescribir como

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \text{ para } i = 1, 2, \ldots, n. \tag{1.3}$$

Si consideramos la suma de los cuadrados de los errores aleatorios, es decir, el cuadrado de la diferencia entre las observaciones con la recta de regresión

$$L = \sum_{i=1}^{n} \epsilon^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \tag{1.4}$$

Para obtener los estimadores por mínimos cuadrados de β_0 y β_1, $\hat{\beta}_0$ y $\hat{\beta}_1$, es preciso calcular las derivadas parciales con respecto a β_0 y β_1, igualar a cero y resolver el sistema de ecuaciones lineales

resultante:

$$\frac{\partial L}{\partial \beta_0} = 0$$
$$\frac{\partial L}{\partial \beta_1} = 0$$

evaluando en $\hat{\beta}_0$ y $\hat{\beta}_1$, se tiene

$$\begin{aligned} -2\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i\right) &= 0 \\ -2\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i\right) x_i &= 0 \end{aligned}$$

simplificando

$$\begin{aligned} n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_i &= \sum_{i=1}^{n} y_i \\ \hat{\beta}_0 \sum_{i=1}^{n} x_i + \hat{\beta}_1 \sum_{i=1}^{n} x_i^2 &= \sum_{i=1}^{n} x_i y_i \end{aligned}$$

Las ecuaciones anteriores se les denominan *ecuaciones normales de mínimos cuadrados* con solución

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x} \tag{1.5}$$
$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right)\left(\sum_{i=1}^{n} x_i\right)}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2} \tag{1.6}$$

entonces el modelo de regresión lineal simple ajustado es

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \tag{1.7}$$

Se intrduce la siguiente notación

$$S_{xx} = \sum_{i=1}^{n} (x_i - \overline{x})^2 = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2 \tag{1.8}$$
$$S_{xy} = \sum_{i=1}^{n} y_i (x_i - \overline{x}) = \sum_{i=1}^{n} x_i y_i - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right) \tag{1.9}$$

y por tanto

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} \tag{1.10}$$

1.1.1 Propiedades de los Estimadores $\hat{\beta}_0$ y $\hat{\beta}_1$

Las propiedades estadísticas de los estimadores de mínimos cuadrados son útiles para evaluar la suficiencia del modelo. Dado que $\hat{\beta}_0$ y $\hat{\beta}_1$ son combinaciones lineales de las variables aleatorias y_i, también resultan ser variables aleatorias. A saber

$$\begin{aligned} E\left(\hat{\beta}_1\right) &= E\left(\frac{S_{xy}}{S_{xx}}\right) = \frac{1}{S_{xx}} E\left(\sum_{i=1}^{n} y_i \left(x_i - \overline{x}\right)\right) \\ &= \frac{1}{S_{xx}} E\left(\sum_{i=1}^{n} \left(\beta_0 + \beta_1 x_i + \epsilon_i\right)\left(x_i - \overline{x}\right)\right) \\ &= \frac{1}{S_{xx}}\left[\beta_0 E\left(\sum_{k=1}^{n}\left(x_k - \overline{x}\right)\right) + E\left(\beta_1 \sum_{k=1}^{n} x_k \left(x_k - \overline{x}\right)\right)\right. \\ &+ \left. E\left(\sum_{k=1}^{n} \epsilon_k \left(x_k - \overline{x}\right)\right)\right] = \frac{1}{S_{xx}} \beta_1 S_{xx} = \beta_1, \end{aligned}$$

por lo tanto

$$E\left(\hat{\beta}_1\right) = \beta_1. \tag{1.11}$$

Es decir, $\hat{\beta}_1$ es un estimador insesgado. Ahora calculemos la varianza:

$$\begin{aligned} V\left(\hat{\beta}_1\right) &= V\left(\frac{S_{xy}}{S_{xx}}\right) = \frac{1}{S_{xx}^2} V\left(\sum_{k=1}^{n} y_k \left(x_k - \overline{x}\right)\right) \\ &= \frac{1}{S_{xx}^2} \sum_{k=1}^{n} V\left(y_k \left(x_k - \overline{x}\right)\right) = \frac{1}{S_{xx}^2} \sum_{k=1}^{n} \sigma^2 \left(x_k - \overline{x}\right)^2 \\ &= \frac{\sigma^2}{S_{xx}^2} \sum_{k=1}^{n} \left(x_k - \overline{x}\right)^2 = \frac{\sigma^2}{S_{xx}} \end{aligned}$$

por lo tanto

$$V\left(\hat{\beta}_1\right) = \frac{\sigma^2}{S_{xx}} \tag{1.12}$$

Se tiene la siguiente proposición

Proposición 1.

$$\begin{aligned} E\left(\hat{\beta}_0\right) &= \beta_0, \\ V\left(\hat{\beta}_0\right) &= \sigma^2\left(\frac{1}{n} + \frac{\overline{x}^2}{S_{xx}}\right), \\ Cov\left(\hat{\beta}_0, \hat{\beta}_1\right) &= -\frac{\sigma^2 \overline{x}}{S_{xx}}. \end{aligned}$$

Para estimar σ^2 es preciso definir la diferencia entre la observación y_k, y el valor predecido $\hat{y}_k$, es decir

$$e_k = y_k - \hat{y}_k, \text{ se le denomina } \textbf{residuo}.$$

La suma de los cuadrados de los errores de los reisduos, *suma de cuadrados del error*

$$SC_E = \sum_{k=1}^{n} e_k^2 = \sum_{k=1}^{n} \left(y_k - \hat{y}_k\right)^2, \tag{1.13}$$

sustituyendo $\hat{y}_k = \hat{\beta}_0 + \hat{\beta}_1 x_k$ se obtiene

$$\begin{aligned} SC_E &= \sum_{k=1}^{n} y_k^2 - n\overline{y}^2 - \hat{\beta}_1 S_{xy} = S_{yy} - \hat{\beta}_1 S_{xy}, \\ E\left(SC_E\right) &= \left(n-2\right)\sigma^2, \text{ por lo tanto} \\ \hat{\sigma}^2 &= \frac{SC_E}{n-2} = MC_E \text{ es un estimador insesgado de } \sigma^2. \end{aligned}$$

1.1.2 Prueba de Hipótesis en RLS

Para evaluar la suficiencia del modelo de regresión lineal simple, es necesario lleva a cabo una prueba de hipótesis respecto de los parámetros del modelo así como de la construcción de intervalos de confianza. Para poder realizar la prueba de hipótesis sobre la pendiente y la ordenada al orígen de la recta de regresión es necesario hacer el supuesto de que el error ϵ_i se distribuye normalmente, es decir $\epsilon_i \sim N\left(0, \sigma^2\right)$. Suponga que se desea probar la hipótesis de que la pendiente es igual a una constante, $\beta_{0,1}$ las hipótesis Nula y Alternativa son:

H_0: $\beta_1 = \beta_{1,0}$,

H_1: $\beta_1 \neq \beta_{1,0}$.

donde dado que las $\epsilon_i \sim N\left(0, \sigma^2\right)$, se tiene que y_i son variables aleatorias normales $N\left(\beta_0 + \beta_1 x_1, \sigma^2\right)$.

De las ecuaciones (1.5) se desprende que $\hat{\beta}_1$ es combinación lineal de variables aleatorias normales independientes, es decir, $\hat{\beta}_1 \sim N\left(\beta_1, \sigma^2/S_{xx}\right)$, recordar las ecuaciones (1.11) y (1.12). Entonces se tiene que el estadístico de prueba apropiado es

$$t_0 = \frac{\hat{\beta}_1 - \hat{\beta}_{1,0}}{\sqrt{MC_E/S_{xx}}}, \tag{1.14}$$

que se distribuye t con $n-2$ grados de libertad bajo $H_0 : \beta_1 = \beta_{1,0}$. Se rechaza H_0 si

$$t_0| > t_{\alpha/2, n-2}. \tag{1.15}$$

Para β_0 se puede proceder de manera análoga para

$H_0 : \beta_0 = \beta_{0,0}$,

$H_1 : \beta_0 \neq \beta_{0,0}$,

con $\hat{\beta}_0 \sim N\left(\beta_0, \sigma^2\left(\frac{1}{n} + \frac{\overline{x}^2}{S_{xx}}\right)\right)$, por lo tanto

$$t_0 = \frac{\hat{\beta}_0 - \beta_{0,0}}{MC_E\left[\frac{1}{n} + \frac{\overline{x}^2}{S_{xx}}\right]}, \tag{1.16}$$

con el que rechazamos la hipótesis nula si

$$t_0| > t_{\alpha/2,n-2}. \tag{1.17}$$

No rechazar $H_0 : \beta_1 = 0$ es equivalente a decir que no hay relación lineal entre x y y. Alternativamente, si $H_0 : \beta_1 = 0$ se rechaza, esto implica que x explica la variabilidad de y, es decir, podría significar que la línea recta esel modelo adecuado. El procedimiento de prueba para $H_0 : \beta_1 = 0$ puede realizarse de la siguiente manera:

$$\begin{aligned}
S_{yy} &= \sum_{k=1}^{n} (y_k - \overline{y})^2 = \sum_{k=1}^{n} (\hat{y}_k - \overline{y})^2 + \sum_{k=1}^{n} (y_k - \hat{y}_k)^2 \\
&= \sum_{k=1}^{n} (y_k - \hat{y}_k + \hat{y}_k - \overline{y})^2 = \sum_{k=1}^{n} [(\hat{y}_k - \overline{y}) + (y_k - \hat{y}_k)]^2 \\
&= \sum_{k=1}^{n} \left[(\hat{y}_k - \overline{y})^2 + 2(\hat{y}_k - \overline{y})(y_k - \hat{y}_k) + (y_k - \hat{y}_k)^2\right] \\
&= \sum_{k=1}^{n} (\hat{y}_k - \overline{y})^2 + 2\sum_{k=1}^{n} (\hat{y}_k - \overline{y})(y_k - \hat{y}_k) + \sum_{k=1}^{n} (y_k - \hat{y}_k)^2 \\
&\quad \sum_{k=1}^{n} (\hat{y}_k - \overline{y})(y_k - \hat{y}_k) = \sum_{k=1}^{n} \hat{y}_k (y_k - \hat{y}_k) - \sum_{k=1}^{n} \overline{y}(y_k - \hat{y}_k) \\
&= \sum_{k=1}^{n} \hat{y}_k (y_k - \hat{y}_k) - \overline{y}\sum_{k=1}^{n} (y_k - \hat{y}_k) \\
&= \sum_{k=1}^{n} \left(\hat{\beta}_0 + \hat{\beta}_1 x_k\right)\left(y_k - \hat{\beta}_0 - \hat{\beta}_1 x_k\right) - \overline{y}\sum_{k=1}^{n} \left(y_k - \hat{\beta}_0 - \hat{\beta}_1 x_k\right) \\
&= \sum_{k=1}^{n} \hat{\beta}_0 \left(y_k - \hat{\beta}_0 - \hat{\beta}_1 x_k\right) + \sum_{k=1}^{n} \hat{\beta}_1 x_k \left(y_k - \hat{\beta}_0 - \hat{\beta}_1 x_k\right) - \overline{y}\sum_{k=1}^{n} \left(y_k - \hat{\beta}_0 - \hat{\beta}_1 x_k\right) \\
&= \hat{\beta}_0 \sum_{k=1}^{n} \left(y_k - \hat{\beta}_0 - \hat{\beta}_1 x_k\right) + \hat{\beta}_1 \sum_{k=1}^{n} x_k \left(y_k - \hat{\beta}_0 - \hat{\beta}_1 x_k\right) - \overline{y}\sum_{k=1}^{n} \left(y_k - \hat{\beta}_0 - \hat{\beta}_1 x_k\right) \\
&= 0 + 0 + 0 = 0.
\end{aligned}$$

Por lo tanto, efectivamente se tiene

$$S_{yy} = \sum_{k=1}^{n} (\hat{y}_k - \overline{y})^2 + \sum_{k=1}^{n} (y_k - \hat{y}_k)^2, \tag{1.18}$$

donde se hacen las definiciones

$$SC_E = \sum_{k=1}^{n} (\hat{y}_k - \overline{y})^2 \cdots \text{Suma de Cuadrados del Error} \tag{1.19}$$

$$SC_R = \sum_{k=1}^{n} (y_k - \hat{y}_k)^2 \cdots \text{ Suma de Regresión de Cuadrados} \tag{1.20}$$

Por lo tanto la ecuación (1.18) se puede reescribir como

$$S_{yy} = SC_R + SC_E, \tag{1.21}$$

recordemos que $SC_E = S_{yy} - \hat{\beta}_1 S_{xy}$

$$\begin{aligned} S_{yy} &= SC_R + \left(S_{yy} - \hat{\beta}_1 S_{xy}\right) \\ S_{xy} &= \frac{1}{\hat{\beta}_1} SC_R, \end{aligned}$$

entonces S_{xy} tiene $n-1$ grados de libertad y SC_R y SC_E tienen 1 y $n-2$ grados de libertad respectivamente.

Proposición 2.

$$E\left(SC_R\right) = \sigma^2 + \beta_1 S_{xx}, \tag{1.22}$$

además, SC_E y SC_R son independientes.

Recordemos que $\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}$. Para $H_0 : \beta_1 = 0$ verdadera,

$$F_0 = \frac{SC_R/1}{SC_E/(n-2)} = \frac{MC_R}{MC_E}$$

se distribuye $F_{1,n-2}$, y se rechazaría H_0 si $F_0 > F_{\alpha,1,n-2}$. El procedimiento de prueba de hipótesis puede presentarse como la tabla de análisis de varianza 1.1:

Table 1.1: Tabla de análisis de varianza

Fuente de variación	Suma de Cuadrados	Grados de Libertad	Media Cuadrática	F_0
Regresión	SC_R	1	MC_R	MC_R/MC_E
Error Residual	SC_E	$n-2$	MC_E	
Total	S_{yy}	$n-1$		

La prueba para la significación de la regresión puede desarrollarse basándose en la expresión (1.14), con $\hat{\beta}_{1,0} = 0$, es decir

$$t_0 = \frac{\hat{\beta}_1}{\sqrt{MC_E/S_{xx}}}. \tag{1.23}$$

Elevando al cuadrado ambos términos:

$$t_0^2 = \frac{\hat{\beta}_1^2 S_{xx}}{MC_E} = \frac{\hat{\beta}_1 S_{xy}}{MC_E} = \frac{MC_R}{MC_E}$$

Observar que $t_0^2 = F_0$, por tanto la prueba que se utiliza para t_0 es la misma que para F_0.

1.1.3 Estimación de Intervalos en RLS

Además de la estimación puntual para los parámetros β_1 y β_0, es posible obtener estimaciones del intervalo de confianza de estos parámetros. El ancho de estos intervalos de confianza es una medida de la calidad total de la recta de regresión. Si los ϵ_k se distribuyen normal e independientemente, entonces

$$\frac{\left(\hat{\beta}_1-\beta_1\right)}{\sqrt{\frac{MC_E}{S_{xx}}}} \quad y \quad \frac{\left(\hat{\beta}_0-\beta_0\right)}{\sqrt{MC_E\left(\frac{1}{n}+\frac{\overline{x}^2}{S_{xx}}\right)}}$$

se distribuyen t con $n-2$ grados de libertad. Por tanto un intervalo de confianza de $100\left(1-\alpha\right)\%$ para β_1 está dado por

$$\hat{\beta}_1 - t_{\alpha/2,n-2}\sqrt{\frac{MC_E}{S_{xx}}} \leq \beta_1 \leq \hat{\beta}_1 + t_{\alpha/2,n-2}\sqrt{\frac{MC_E}{S_{xx}}}. \tag{1.24}$$

De igual manera, para β_0 un intervalo de confianza al $100\left(1-\alpha\right)\%$ es

$$\hat{\beta}_0 - t_{\alpha/2,n-2}\sqrt{MC_E\left(\frac{1}{n}+\frac{\overline{x}^2}{S_{xx}}\right)} \leq \beta_0 \leq \hat{\beta}_0 + t_{\alpha/2,n-2}\sqrt{MC_E\left(\frac{1}{n}+\frac{\overline{x}^2}{S_{xx}}\right)} \tag{1.25}$$

1.1.4 Predicción

Supongamos que se tiene un valor x_0 de interés, entonces la estimación puntual de este nuevo valor

$$\hat{y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_0. \tag{1.26}$$

Esta nueva observación es independiente de las utilizadas para obtener el modelo de regresión, por tanto, el intervalo en torno a la recta de regresión es inapropiado, puesto que se basa únicamente en los datos empleados para ajustar el modelo de regresión. El intervalo de confianza en torno a la recta de regresión se refiere a la respuesta media verdadera $x = x_0$, no a observaciones futuras. Sea y_0 la observación futura en $x = x_0$, y sea $\hat{y}_0$ dada en la ecuación anterior, el estimador de y_0. Si se define la variable aleatoria

$$w = y_0 - \hat{y}_0,$$

esta se distribuye normalmente con media cero y varianza

$$V\left(w\right) = \sigma^2\left[1+\frac{1}{n}+\frac{\left(x-x_0\right)^2}{S_{xx}}\right]$$

dado que y_0 es independiente de $\hat{y}_0$, por lo tanto el intervalo de predicción al nivel α para futuras observaciones x_0 es

$$\hat{y}_0 - t_{\alpha/2,n-2}\sqrt{MC_E\left[1+\frac{1}{n}+\frac{\left(x-x_0\right)^2}{S_{xx}}\right]} \leq y_0$$
$$\leq \hat{y}_0 + t_{\alpha/2,n-2}\sqrt{MC_E\left[1+\frac{1}{n}+\frac{\left(x-x_0\right)^2}{S_{xx}}\right]}.$$

Es común encontrar que el modelo ajustado no satisface totalmente el modelo necesario para los datos, en este caso es preciso saber qué tan bueno es el modelo propuesto. Para esto se propone la siguiente prueba de hipótesis:

H_0 : El modelo propuesto se ajusta adecuademente a los datos.

H_1 : El modelo NO se ajusta a los datos.

La prueba implica dividir la suma de cuadrados del eror o del residuo en las siguientes dos componentes:

$$SC_E = SC_{EP} + SC_{FDA},$$

donde SC_{EP} es la suma de cuadrados atribuibles al error puro, y SC_{FDA} es la suma de cuadrados atribuible a la falta de ajuste del modelo. La cantidad

$$R^2 = \frac{SC_R}{S_{yy}} = 1 - \frac{SC_E}{S_{yy}}, \tag{1.27}$$

se denomina coeficiente de determinación y se utiliza para saber si el modelo de regresión es suficiente o no. Se puede demostrar que $0 \leq R^2 \leq 1$, una manera de interpretar este valor es que si $R^2 = k$, entonces el modelo de regresión explica el $k * 100\%$ de la variabilidad en los datos. R^2

- No mide la magnitud de la pendiente de la recta de regresión.
- Un valor grande de R^2 no implica una pendiente empinada.
- No mide la suficiencia del modelo.
- Valores grandes de R^2 no implican necesariamente que el modelo de regresión proporcionará predicciones precisas para futuras observaciones.

Chapter 2

Regresión Logística

2.1 Regresión múltiple

La regresión logística es una técnica de modelado estadístico utilizada para predecir la probabilidad de un evento binario (es decir, un evento que tiene dos posibles resultados) en función de una o más variables independientes. Un modelo de regresión logística describe cómo una variable dependiente binaria Y (que puede tomar los valores 0 o 1) está relacionada con una o más variables independientes $X_1, X_2, \ldots, X_n$. A diferencia de la regresión lineal, que predice un valor continuo, la regresión logística predice una probabilidad que puede ser interpretada como la probabilidad de que $Y = 1$ dado un conjunto de valores para $X_1, X_2, \ldots, X_n$, la regresión logística está diseñada para manejar situaciones donde la respuesta es categórica. En su forma más común, la regresión logística binaria, el modelo predice la probabilidad de que un evento ocurra en función de una o más variables independientes. Este tipo de regresión toma la forma de un modelo no lineal, debido a la naturaleza discreta de la variable dependiente. La regresión lineal busca modelar la relación entre una variable dependiente continua Y y una o más variables independientes $X_1, X_2, \ldots, X_n$ mediante una ecuación de la forma:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon, \tag{2.1}$$

donde $\beta_0, \beta_1, \ldots, \beta_n$ son los coeficientes del modelo y ϵ es el término de error. La ecuación de la regresión logística es:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n, \tag{2.2}$$

donde p es la probabilidad de que $Y = 1$. La función logística es:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}}. \tag{2.3}$$

La regresión lineal es utilizada para predecir el valor de una variable dependiente continua en función de una o más variables independientes. El modelo de regresión lineal tiene la forma:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon, \tag{2.4}$$

donde:

a) Y es la variable dependiente.

b) β_0 es la intersección con el eje Y o término constante.

c) $\beta_1, \beta_2, \ldots, \beta_n$ son los coeficientes que representan la relación entre las variables independientes y la variable dependiente.

d) $X_1, X_2, \ldots, X_n$ son las variables independientes.

e) ϵ es el término de error, que representa la desviación de los datos observados de los valores predichos por el modelo.

El objetivo de la regresión lineal es encontrar los valores de los coeficientes $\beta_0, \beta_1, \ldots, \beta_n$ que minimicen la suma de los cuadrados de las diferencias entre los valores observados y los valores predichos. Este método se conoce como mínimos cuadrados ordinarios (OLS, por sus siglas en inglés). La función de costo a minimizar es:

$$J(\beta_0, \beta_1, \ldots, \beta_n) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{2.5}$$

donde:

a) y_i es el valor observado de la variable dependiente para la i-ésima observación.

b) $\hat{y}_i$ es el valor predicho por el modelo para la i-ésima observación, dado por:

$$\hat{y}_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_n x_{in}. \tag{2.6}$$

Para encontrar los valores óptimos de los coeficientes, se toman las derivadas parciales de la función de costo con respecto a cada coeficiente y se igualan a cero:

$$\frac{\partial J}{\partial \beta_j} = 0 \quad \text{para } j = 0, 1, \ldots, n. \tag{2.7}$$

Resolviendo este sistema de ecuaciones, se obtienen los valores de los coeficientes que minimizan la función de costo. La deducción de la fórmula de la regresión logística comienza con la necesidad de modelar la probabilidad de un evento binario. Queremos encontrar una función que relacione las variables independientes con la probabilidad de que la variable dependiente tome el valor 1. La probabilidad de que el evento ocurra, $P(Y = 1)$, se denota como p. La probabilidad de que el evento no ocurra, $P(Y = 0)$, es $1 - p$. Los **odds** (chances) de que ocurra el evento se definen como:

$$\text{odds} = \frac{p}{1-p}. \tag{2.8}$$

Los odds indican cuántas veces más probable es que ocurra el evento frente a que no ocurra. Para simplificar el modelado de los *odds*, se aplica el logaritmo natural, obteniendo la función **logit**:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right). \tag{2.9}$$

La transformación logit es útil porque convierte el rango de la probabilidad (0, 1) al rango de números reales $(-\infty, \infty)$. La idea clave de la regresión logística es modelar la transformación logit de la probabilidad como una combinación lineal de las variables independientes:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n. \tag{2.10}$$

Aquí, β_0 es el término constante y $\beta_1, \beta_2, \ldots, \beta_n$ son los coeficientes asociados con las variables independientes $X_1, X_2, \ldots, X_n$. Para expresar p en función de una combinación lineal de las variables independientes, invertimos la transformación logit. Partimos de la ecuación 2.10, aplicando la función exponencial en ambos lados:

$$\frac{p}{1-p} = e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n}. \tag{2.11}$$

Despejando p:

$$p = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n}}. \tag{2.12}$$

La expresión final que obtenemos es conocida como la **función logística**:

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n)}}. \tag{2.13}$$

Esta función describe cómo las variables independientes se relacionan con la probabilidad de que el evento de interés ocurra. Los coeficientes $\beta_0, \beta_1, \ldots, \beta_n$ se estiman a partir de los datos utilizando el método de máxima verosimilitud.

2.2 Método de Máxima Verosimilitud

Para estimar los coeficientes $\beta_0, \beta_1, \ldots, \beta_n$ en la regresión logística, utilizamos el método de máxima verosimilitud. La idea es encontrar los valores de los coeficientes que maximicen la probabilidad de observar los datos dados. Esta probabilidad se expresa mediante la función de verosimilitud L. La función de verosimilitud $L(\beta_0, \beta_1, \ldots, \beta_n)$ para un conjunto de n observaciones se define como el producto de las probabilidades de las observaciones dadas las variables independientes:

$$L(\beta_0, \beta_1, \ldots, \beta_n) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i} \tag{2.14}$$

donde:

- p_i es la probabilidad predicha de que $Y_i = 1$,
- y_i es el valor observado de la variable dependiente para la i-ésima observación.

Trabajar directamente con esta función de verosimilitud puede ser complicado debido al producto de muchas probabilidades, especialmente si n es grande. Para simplificar los cálculos, se utiliza el logaritmo de la función de verosimilitud, conocido como la función de *log-verosimilitud*. El uso del logaritmo simplifica significativamente la diferenciación y maximización de la función. La función de log-verosimilitud se define como:

$$\log L(\beta_0, \beta_1, \ldots, \beta_n) = \sum_{i=1}^{n} \left[y_i \log(p_i) + (1-y_i)\log(1-p_i) \right]. \tag{2.15}$$

Aquí, log representa el logaritmo natural. Esta transformación es válida porque el logaritmo es una función monótona creciente, lo que significa que maximizar la log-verosimilitud es equivalente a maximizar la verosimilitud original. En la regresión logística, la probabilidad p_i está dada por la función logística:

$$p_i = \frac{1}{1+e^{-(\beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}+\ldots+\beta_n X_{in})}}. \tag{2.16}$$

Sustituyendo esta expresión en la función de log-verosimilitud, obtenemos:

$$\begin{aligned} \log L(\beta_0, \beta_1, \ldots, \beta_n) \quad &= \textstyle\sum_{i=1}^{n} \left[y_i \log\left(\frac{1}{1+e^{-(\beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}+\ldots+\beta_n X_{in})}} \right) + \right. \\ &\quad \left. (1-y_i)\log\left(1 - \frac{1}{1+e^{-(\beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}+\ldots+\beta_n X_{in})}} \right) \right]. \end{aligned}$$

Simplificando esta expresión, notamos que:

$$\log\left(\frac{1}{1+e^{-z}}\right) = -\log(1+e^{-z}),$$

y

$$\log\left(1-\frac{1}{1+e^{-z}}\right) = \log\left(\frac{e^{-z}}{1+e^{-z}}\right) = -z-\log(1+e^{-z}).$$

Aplicando estas identidades, la función de log-verosimilitud se convierte en:

$$\begin{aligned}\log L(\beta_0,\beta_1,\ldots,\beta_n) &= \sum_{i=1}^{n}\Big[y_i(-\log(1+e^{-(\beta_0+\beta_1X_{i1}+\beta_2X_{i2}+\ldots+\beta_nX_{in})})) \\ &+ (1-y_i)\left(-(\beta_0+\beta_1X_{i1}+\beta_2X_{i2}+\ldots+\beta_nX_{in})\right) \\ &- (1-y_i)\log(1+e^{-(\beta_0+\beta_1X_{i1}+\beta_2X_{i2}+\ldots+\beta_nX_{in})})\Big].\end{aligned}$$

Simplificando aún más, obtenemos:

$$\begin{aligned}\log L(\beta_0,\beta_1,\ldots,\beta_n) &= \sum_{i=1}^{n}[y_i(\beta_0+\beta_1X_{i1}+\beta_2X_{i2}+\ldots+\beta_nX_{in}) \\ &- \log(1+e^{\beta_0+\beta_1X_{i1}+\beta_2X_{i2}+\ldots+\beta_nX_{in}})\Big].\end{aligned}$$

Para simplificar aún más la notación, podemos utilizar notación matricial. Definimos la matriz $\mathbf{X}$ de tamaño $n\times(k+1)$ y el vector de coeficientes $\boldsymbol{\beta}$ de tamaño $(k+1)\times 1$ como sigue:

$$\mathbf{X} = \begin{bmatrix}1 & x_{11} & x_{12} & \ldots & x_{1k}\\ 1 & x_{21} & x_{22} & \ldots & x_{2k}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk}\end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix}\beta_0\\ \beta_1\\ \beta_2\\ \vdots\\ \beta_k\end{bmatrix}. \tag{2.17}$$

Entonces, la expresión para la función de log-verosimilitud es:

$$\log L(\boldsymbol{\beta}) = \sum_{i=1}^{n}\left[y_i(\mathbf{X}_i\boldsymbol{\beta}) - \log(1+e^{\mathbf{X}_i\boldsymbol{\beta}})\right], \tag{2.18}$$

donde $\mathbf{X}_i$ es la i-ésima fila de la matriz $\mathbf{X}$. Esta notación matricial simplifica la implementación y la derivación de los estimadores de los coeficientes en la regresión logística. Utilizando métodos numéricos, como el algoritmo de *Newton-Raphson*, se pueden encontrar los coeficientes que maximizan la función de log-verosimilitud. Para maximizar la función de log-verosimilitud, derivamos esta función con respecto a cada uno de los coeficientes β_j y encontramos los puntos críticos. La derivada parcial de la función de log-verosimilitud con respecto a β_j es:

$$\frac{\partial \log L(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{n}\left[y_iX_{ij} - \frac{X_{ij}e^{\mathbf{X}_i\boldsymbol{\beta}}}{1+e^{\mathbf{X}_i\boldsymbol{\beta}}}\right]. \tag{2.19}$$

Simplificando, esta derivada se puede expresar como:

$$\frac{\partial \log L(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{n} X_{ij}(y_i - p_i), \text{ donde } p_i = \frac{1}{1 + e^{-\mathbf{X}_i \boldsymbol{\beta}}}. \tag{2.20}$$

Para encontrar los coeficientes que maximizan la log-verosimilitud, resolvemos el sistema de ecuaciones

$$\frac{\partial \log L(\boldsymbol{\beta})}{\partial \beta_j} = 0 \text{ para todos los } j = 0, 1, \ldots, k.$$

Este sistema de ecuaciones no tiene una solución analítica cerrada, por lo que se resuelve numéricamente utilizando métodos iterativos como el algoritmo de Newton-Raphson.

2.3 Método de Newton-Raphson

El método de Newton-Raphson es un algoritmo iterativo que se utiliza para encontrar las raíces de una función. En el contexto de la regresión logística, se utiliza para maximizar la función de log-verosimilitud encontrando los valores de los coeficientes $\beta_0, \beta_1, \ldots, \beta_n$. Este método se basa en una aproximación de segundo orden de la función objetivo. Dado un valor inicial de los coeficientes $\boldsymbol{\beta}^{(0)}$, se actualiza iterativamente el valor de los coeficientes utilizando la fórmula:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left[\mathbf{H}(\boldsymbol{\beta}^{(t)})\right]^{-1} \nabla \log L(\boldsymbol{\beta}^{(t)}), \tag{2.21}$$

donde:

- $\boldsymbol{\beta}^{(t)}$ es el vector de coeficientes en la t-ésima iteración.
- $\nabla \log L(\boldsymbol{\beta}^{(t)})$ es el gradiente de la función de log-verosimilitud con respecto a los coeficientes $\boldsymbol{\beta}$:

  $$\nabla \log L(\boldsymbol{\beta}) = \mathbf{X}^T(\mathbf{y} - \mathbf{p}), \tag{2.22}$$

  donde $\mathbf{y}$ es el vector de valores observados y $\mathbf{p}$ es el vector de probabilidades.
- $\mathbf{H}(\boldsymbol{\beta}^{(t)})$ es la matriz Hessiana (matriz de segundas derivadas) evaluada en $\boldsymbol{\beta}^{(t)}$:

  $$\mathbf{H}(\boldsymbol{\beta}) = -\mathbf{X}^T \mathbf{W} \mathbf{X}, \tag{2.23}$$

  donde $\mathbf{W}$ es una matriz diagonal de pesos con elementos $w_i = p_i(1 - p_i)$.

Algoritmo 1. *El algoritmo Newton-Raphson para la regresión logística se puede resumir en los siguientes pasos:*

1. *Inicializar el vector de coeficientes $\boldsymbol{\beta}^{(0)}$ (por ejemplo, con ceros o valores pequeños aleatorios).*
2. *Calcular el gradiente $\nabla \log L(\boldsymbol{\beta}^{(t)})$ y la matriz Hessiana $\mathbf{H}(\boldsymbol{\beta}^{(t)})$ en la iteración t.*
3. *Actualizar los coeficientes utilizando la fórmula:*

   $$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left[\mathbf{H}(\boldsymbol{\beta}^{(t)})\right]^{-1} \nabla \log L(\boldsymbol{\beta}^{(t)}) \tag{2.24}$$

4. *Repetir los pasos 2 y 3 hasta que la diferencia entre $\boldsymbol{\beta}^{(t+1)}$ y $\boldsymbol{\beta}^{(t)}$ sea menor que un umbral predefinido (criterio de convergencia).*

El método de Newton-Raphson permite encontrar los coeficientes que maximizan la función de log-verosimilitud de manera eficiente. En específico para un conjunto de n observaciones, la función de verosimilitud L se define como el producto de las probabilidades individuales de observar cada dato:

$$L(\beta_0, \beta_1, \ldots, \beta_n) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}, \tag{2.25}$$

donde y_i es el valor observado de la variable dependiente para la i-ésima observación y p_i es la probabilidad predicha de que $Y_i = 1$. Aquí, p_i es dado por la función logística:

$$p_i = \frac{1}{1 + e^{-(\beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}+\ldots+\beta_n X_{in})}}. \tag{2.26}$$

Tomando el logaritmo:

$$\log L(\beta_0, \beta_1, \ldots, \beta_n) = \sum_{i=1}^{n} \left[y_i \log(p_i) + (1-y_i)\log(1-p_i) \right]. \tag{2.27}$$

Sustituyendo p_i:

$$\log L(\beta_0, \beta_1, \ldots, \beta_n) = \sum_{i=1}^{n} \left[y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_n X_{in}) - \log(1 + e^{\beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}+\ldots+\beta_n X_{in}}) \right]. \tag{2.28}$$

Dado que el objetivo es encontrar los valores de $\beta_0, \beta_1, \ldots, \beta_n$ que maximicen la función de log-verosimilitud. Para β_j, la derivada parcial de la función de log-verosimilitud es:

$$\frac{\partial \log L}{\partial \beta_j} = \sum_{i=1}^{n} \left[y_i X_{ij} - \frac{X_{ij} e^{\beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}+\ldots+\beta_n X_{in}}}{1 + e^{\beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}+\ldots+\beta_n X_{in}}} \right]. \tag{2.29}$$

Esto se simplifica a (comparar con la ecuación 2.19):

$$\frac{\partial \log L}{\partial \beta_j} = \sum_{i=1}^{n} X_{ij}(y_i - p_i). \tag{2.30}$$

Para maximizar la log-verosimilitud, resolvemos el sistema de ecuaciones $\frac{\partial \log L}{\partial \beta_j} = 0$ para todos los j de 0 a n, mismo que se resuelve numéricamente utilizando métodos el algoritmo de Newton-Raphson. El método de Newton-Raphson se basa en una aproximación de segundo orden de la función objetivo. Dado un valor inicial de los coeficientes $\beta^{(0)}$, se iterativamente actualiza el valor de los coeficientes utilizando la fórmula:

$$\beta^{(k+1)} = \beta^{(k)} - \left[\mathbf{H}(\beta^{(k)}) \right]^{-1} \mathbf{g}(\beta^{(k)}), \tag{2.31}$$

donde:

- $\beta^{(k)}$ es el vector de coeficientes en la k-ésima iteración.
- $\mathbf{g}(\beta^{(k)})$ es el gradiente (vector de primeras derivadas) evaluado en $\beta^{(k)}$:

$$\mathbf{g}(\beta) = \frac{\partial \log L}{\partial \beta} = \sum_{i=1}^{n} \mathbf{X}_i(y_i - p_i), \tag{2.32}$$

  donde $\mathbf{X}_i$ es el vector de valores de las variables independientes para la i-ésima observación (comparar con ecuación 2.22).

- $\mathbf{H}(\beta^{(k)})$ es la matriz Hessiana (matriz de segundas derivadas) evaluada en $\beta^{(k)}$:

$$\mathbf{H}(\beta) = \frac{\partial^2 \log L}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{n} p_i(1-p_i)\mathbf{X}_i\mathbf{X}_i^T, \tag{2.33}$$

  comparar con ecuación 2.23.

Algoritmo 2. *Los pasos del algoritmo Newton-Raphson para la regresión logística son:*

1. *Inicializar el vector de coeficientes $\beta^{(0)}$ (por ejemplo, con ceros o valores pequeños aleatorios).*
2. *Calcular el gradiente $\mathbf{g}(\beta^{(k)})$ y la matriz Hessiana $\mathbf{H}(\beta^{(k)})$ en la iteración k.*
3. *Actualizar los coeficientes utilizando la fórmula:*

$$\beta^{(k+1)} = \beta^{(k)} - \left[\mathbf{H}(\beta^{(k)})\right]^{-1} \mathbf{g}(\beta^{(k)}). \tag{2.34}$$

4. *Repetir los pasos 2 y 3 hasta que la diferencia entre $\beta^{(k+1)}$ y $\beta^{(k)}$ sea menor que un umbral predefinido (criterio de convergencia).*

Como se puede observar, la diferencia entre el Algoritmo 1 y el Algoritmo 2 son mínimas.

2.4 Notas finales

Nota 1. *En el contexto de la regresión logística, los vectores $X_1, X_2, \ldots, X_n$ representan las variables independientes. Cada X_j es un vector columna que contiene los valores de la variable independiente j para cada una de las n observaciones. Es decir,*

$$X_j = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{bmatrix}. \tag{2.35}$$

Para simplificar la notación y los cálculos, a menudo combinamos todos los vectores de variables independientes en una única matriz de diseño $\mathbf{X}$ de tamaño $n \times (k+1)$, donde n es el número de observaciones y $k+1$ es el número de variables independientes más el término de intercepto. La primera columna de $\mathbf{X}$ corresponde a un vector de unos para el término de intercepto, y las demás columnas corresponden a los valores de las variables independientes:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk}, \end{bmatrix} \tag{2.36}$$

revisar la ecuación 2.17. De esta forma, el modelo logit puede ser escrito de manera compacta utilizando la notación matricial:

$$\mathit{logit}(p) = \log\left(\frac{p}{1-p}\right) = \mathbf{X}\boldsymbol{\beta}, \tag{2.37}$$

donde $\boldsymbol{\beta}$ es el vector de coeficientes:

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix}. \tag{2.38}$$

Así, la probabilidad p se puede expresar como:

$$p = \frac{1}{1 + e^{-\mathbf{X}\boldsymbol{\beta}}}. \tag{2.39}$$

Comparar la ecuación anterior con la ecuación 2.13. Esta notación matricial simplifica la implementación y la derivación de los estimadores de los coeficientes en la regresión logística. Para estimar los coeficientes $\boldsymbol{\beta}$ en la regresión logística, se utiliza el método de máxima verosimilitud. La función de verosimilitud $L(\boldsymbol{\beta})$ se define como el producto de las probabilidades de las observaciones dadas las variables independientes, recordemos la ecuación 2.14:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1 - y_i}, \tag{2.40}$$

donde y_i es el valor observado de la variable dependiente para la i-ésima observación, y p_i es la probabilidad predicha de que $Y_i = 1$. La función de log-verosimilitud, que es más fácil de maximizar, se obtiene tomando el logaritmo natural de la función de verosimilitud (2.18):

$$\log L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right]. \tag{2.41}$$

Sustituyendo $p_i = \frac{1}{1+e^{-\mathbf{X}_i\boldsymbol{\beta}}}$, donde $\mathbf{X}_i$ es la i-ésima fila de la matriz de diseño $\mathbf{X}$, obtenemos:

$$\log L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i (\mathbf{X}_i \boldsymbol{\beta}) - \log(1 + e^{\mathbf{X}_i\boldsymbol{\beta}}) \right]. \tag{2.42}$$

Para encontrar los valores de $\boldsymbol{\beta}$ que maximizan la función de log-verosimilitud, se utiliza un algoritmo iterativo como el método de Newton-Raphson. Este método requiere calcular el gradiente y la matriz Hessiana de la función de log-verosimilitud. El gradiente de la función de log-verosimilitud con respecto a $\boldsymbol{\beta}$ es (2.22 y 2.32):

$$\nabla \log L(\boldsymbol{\beta}) = \mathbf{X}^T (\mathbf{y} - \mathbf{p}), \tag{2.43}$$

donde $\mathbf{y}$ es el vector de valores observados y $\mathbf{p}$ es el vector de probabilidades predichas. La matriz Hessiana de la función de log-verosimilitud es (2.23 y 2.33):

$$\mathbf{H}(\boldsymbol{\beta}) = -\mathbf{X}^T \mathbf{W} \mathbf{X}, \tag{2.44}$$

donde $\mathbf{W}$ es una matriz diagonal de pesos con elementos $w_i = p_i(1 - p_i)$. El método de Newton-Raphson actualiza los coeficientes $\boldsymbol{\beta}$ de la siguiente manera:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - [\mathbf{H}(\boldsymbol{\beta}^{(t)})]^{-1} \nabla \log L(\boldsymbol{\beta}^{(t)}). \tag{2.45}$$

Iterando este proceso hasta que la diferencia entre $\boldsymbol{\beta}^{(t+1)}$ y $\boldsymbol{\beta}^{(t)}$ sea menor que un umbral predefinido (2.21, 2.24, 2.31 y 2.34), se obtienen los estimadores de máxima verosimilitud para los coeficientes de la regresión logística.

Nota 2. ***Selección de Variables***

La selección de variables es el proceso de elegir las variables más relevantes para el modelo. Existen varias técnicas para la selección de variables:

Métodos de Filtrado: *Los métodos de filtrado seleccionan variables basadas en criterios estadísticos, como la correlación o la chi-cuadrado. Algunas técnicas comunes incluyen:*

- ***Análisis de Correlación****: Se seleccionan variables con alta correlación con la variable dependiente y baja correlación entre ellas.*
- ***Pruebas de Chi-cuadrado****: Se utilizan para variables categóricas para determinar la asociación entre la variable independiente y la variable dependiente.*

Métodos de Wrapper: *Los métodos de wrapper evalúan múltiples combinaciones de variables y seleccionan la combinación que optimiza el rendimiento del modelo. Ejemplos incluyen:*

- ***Selección hacia Adelante****: Comienza con un modelo vacío y agrega variables una por una, seleccionando la variable que mejora más el modelo en cada paso.*
- ***Selección hacia Atrás****: Comienza con todas las variables y elimina una por una, removiendo la variable que tiene el menor impacto en el modelo en cada paso.*
- ***Selección Paso a Paso****: Combina la selección hacia adelante y hacia atrás, agregando y eliminando variables según sea necesario.*

Métodos Basados en Modelos: *Los métodos basados en modelos utilizan técnicas de regularización como Lasso y Ridge para seleccionar variables. Estas técnicas añaden un término de penalización a la función de costo para evitar el sobreajuste.*

Regresión Lasso: *La regresión Lasso (Least Absolute Shrinkage and Selection Operator) añade una penalización L_1 a la función de costo:*

$$J(\beta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|,$$

donde λ es el parámetro de regularización que controla la cantidad de penalización.

Regresión Ridge: *La regresión Ridge añade una penalización L_2 a la función de costo:*

$$J(\beta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2,$$

donde λ es el parámetro de regularización.

Nota 3. *Evaluar la calidad y el rendimiento de un modelo de regresión logística es crucial para asegurar que las predicciones sean precisas y útiles. Este capítulo se centra en las técnicas y métricas utilizadas para evaluar modelos de clasificación binaria, así como en la validación cruzada, una técnica para evaluar la generalización del modelo. Las métricas de evaluación permiten cuantificar la precisión y el rendimiento de un modelo. Algunas de las métricas más comunes incluyen:*

Curva ROC y AUC: *La curva ROC (Receiver Operating Characteristic) es una representación gráfica de la sensibilidad (verdaderos positivos) frente a 1 - especificidad (falsos positivos). El área bajo la curva (AUC) mide la capacidad del modelo para distinguir entre las clases.*

$$\begin{aligned} \textit{Sensibilidad} &= \frac{TP}{TP + FN} \\ \textit{Especificidad} &= \frac{TN}{TN + FP} \end{aligned}$$

Matriz de Confusión: *La matriz de confusión es una tabla que muestra el rendimiento del modelo comparando las predicciones con los valores reales. Los términos incluyen:*

- ***Verdaderos Positivos (TP)**: Predicciones correctas de la clase positiva.*
- ***Falsos Positivos (FP)**: Predicciones incorrectas de la clase positiva.*
- ***Verdaderos Negativos (TN)**: Predicciones correctas de la clase negativa.*
- ***Falsos Negativos (FN)**: Predicciones incorrectas de la clase negativa.*

	Predicción Positiva	**Predicción Negativa**
Real Positiva	TP	FN
Real Negativa	FP	TN

Table 2.1: Matriz de Confusión

***Precisión, Recall y F1-Score:** se define como*

$$\begin{aligned} \textit{Precisión} &= \frac{TP}{TP+FP} \\ \textit{Recall} &= \frac{TP}{TP+FN} \\ \textit{F1-Score} &= 2 \cdot \frac{\textit{Precisión} \cdot \textit{Recall}}{\textit{Precisión} + \textit{Recall}}. \end{aligned}$$

***Log-Loss:** La pérdida logarítmica (Log-Loss) mide la precisión de las probabilidades predichas. La fórmula es:*

$$\textit{Log-Loss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right],$$

donde y_i son los valores reales y p_i son las probabilidades predichas.

***Validación Cruzada:** La validación cruzada es una técnica para evaluar la capacidad de generalización de un modelo. Existen varios tipos de validación cruzada:*

***K-Fold Cross-Validation:** En K-Fold Cross-Validation, los datos se dividen en K subconjuntos. El modelo se entrena K veces, cada vez utilizando K-1 subconjuntos para el entrenamiento y el subconjunto restante para la validación.*

$$\textit{Error Medio} = \frac{1}{K}\sum_{k=1}^{K} \textit{Error}_k.$$

***Leave-One-Out Cross-Validation (LOOCV):** En LOOCV, cada observación se usa una vez como conjunto de validación y las restantes como conjunto de entrenamiento. Este método es computacionalmente costoso pero útil para conjuntos de datos pequeños.*

***Ajuste y Sobreajuste del Modelo:** El ajuste adecuado del modelo es crucial para evitar el sobreajuste (overfitting) y el subajuste (underfitting).*

***Sobreajuste:** El sobreajuste ocurre cuando un modelo se ajusta demasiado bien a los datos de entrenamiento, capturando ruido y patrones irrelevantes. Los síntomas incluyen una alta precisión en el entrenamiento y baja precisión en la validación.*

***Subajuste:** El subajuste ocurre cuando un modelo no captura los patrones subyacentes de los datos. Los síntomas incluyen baja precisión tanto en el entrenamiento como en la validación.*

***Regularización:** La regularización es una técnica para prevenir el sobreajuste añadiendo un término de penalización a la función de costo. Las técnicas comunes incluyen:*

- ***Regresión Lasso (L1)***
- ***Regresión Ridge (L2)***

Nota 4. ***Interpretación de los Resultados:*** *Interpretar correctamente los resultados de un modelo de regresión logística es esencial para tomar decisiones informadas. Este capítulo se centra en la interpretación de los coeficientes del modelo, las odds ratios, los intervalos de confianza y la significancia estadística.*

Coeficientes de Regresión Logística: *Los coeficientes de regresión logística representan la relación entre las variables independientes y la variable dependiente en términos de log-odds. Cada coeficiente β_j en el modelo de regresión logística se interpreta como el cambio en el log-odds de la variable dependiente por unidad de cambio en la variable independiente X_j.*

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

Signo de los Coeficientes:

- ***Coeficiente Positivo****: Un coeficiente positivo indica que un aumento en la variable independiente está asociado con un aumento en el log-odds de la variable dependiente.*
- ***Coeficiente Negativo****: Un coeficiente negativo indica que un aumento en la variable independiente está asociado con una disminución en el log-odds de la variable dependiente.*

Odds Ratios: *Las odds ratios proporcionan una interpretación más intuitiva de los coeficientes de regresión logística. La odds ratio para una variable independiente X_j se calcula como e^{β_j}. Recordemos que*

$$OR_j = e^{\beta_j}$$

Interpretación de las Odds Ratios:

- *$OR > 1$: Un OR mayor que 1 indica que un aumento en la variable independiente está asociado con un aumento en las odds de la variable dependiente.*
- *$OR < 1$: Un OR menor que 1 indica que un aumento en la variable independiente está asociado con una disminución en las odds de la variable dependiente.*
- *$OR = 1$: Un OR igual a 1 indica que la variable independiente no tiene efecto sobre las odds de la variable dependiente.*

Intervalos de Confianza: *Los intervalos de confianza proporcionan una medida de la incertidumbre asociada con los estimadores de los coeficientes. Un intervalo de confianza del 95% para un coeficiente β_j indica que, en el 95% de las muestras, el intervalo contendrá el valor verdadero de β_j.*

Para calcular un intervalo de confianza del 95% para un coeficiente β_j,

$$\beta_j \pm 1.96 \cdot SE(\beta_j),$$

donde $SE(\beta_j)$ es el error estándar de β_j.

Significancia Estadística: *La significancia estadística se utiliza para determinar si los coeficientes del modelo son significativamente diferentes de cero. Esto se evalúa mediante pruebas de hipótesis. Para cada coeficiente β_j, la hipótesis nula H_0 es que $\beta_j = 0$. La hipótesis alternativa H_a es que $\beta_j \neq 0$.*

P-valor: *El p-valor indica la probabilidad de obtener un coeficiente tan extremo como el observado, asumiendo que la hipótesis nula es verdadera. Un p-valor menor que el nivel de significancia α (típicamente 0.05) indica que podemos rechazar la hipótesis nula.*

Bibliography

[1] Darlington RB. *Regression and Linear Models*. Columbus, OH: McGraw-Hill Publishing Company, 1990.

[2] Tabachnick BG, Fidell LS. *Using Multivariate Statistics*. 5th ed. Boston, MA: Pearson Education, Inc., 2007.

[3] Hosmer DW, Lemeshow SL. *Applied Logistic Regression*. 2nd ed. Hoboken, NJ: Wiley-Interscience, 2000.

[4] Campbell DT, Stanley JC. *Experimental and Quasi-experimental Designs for Research*. Boston, MA: Houghton Mifflin Co., 1963.

[5] Stokes ME, Davis CS, Koch GG. *Categorical Data Analysis Using the SAS System*. 2nd ed. Cary, NC: SAS Institute, Inc., 2000.

[6] Newgard CD, Hedges JR, Arthur M, Mullins RJ. Advanced statistics: the propensity score—a method for estimating treatment effect in observational research. *Acad Emerg Med*. 2004; **11**:953–961.

[7] Newgard CD, Haukoos JS. Advanced statistics: missing data in clinical research—part 2: multiple imputation. *Acad Emerg Med*. 2007; **14**:669–678.

[8] Allison PD. *Logistic Regression Using the SAS System: Theory and Application*. Cary, NC: SAS Institute, Inc., 1999.

[9] Peduzzi P, Concato J, Kemper E, Holford TR, Feinstein AR. A simulation study of the number of events per variable in logistic regression analysis. *J Clin Epidemiol*. 1996; **49**:1373–1379.

[10] Agresti A. *An Introduction to Categorical Data Analysis*. Hoboken, NJ: Wiley, 2007.

[11] Feinstein AR. *Multivariable Analysis: An Introduction*. New Haven, CT: Yale University Press, 1996.

[12] Altman DG, Royston P. What Do We Mean by Validating a Prognostic Model? *Stats Med*. 2000; **19**:453–473.

[13] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*. Montreal, Quebec, Canada, August 20–25, 1995. 1995:1137–1143.

[14] Efron B, Tibshirani R. *An Introduction to the Bootstrap*. New York: Chapman & Hall, 1993.

[15] Miller ME, Hiu SL, Tierney WM. Validation techniques for logistic regression models. *Stat Med*. 1991; **10**:1213–1226.

[16] Hosmer DW, Hosmer T, Le Cessie S, Lemeshow S. A comparison of goodness-of-fit tests for the logistic regression model. *Stat Med*. 1997; **16**:965–980.

[17] Kuss O. Global goodness-of-fit tests in logistic regression with sparse data. *Stat Med*. 2002; **21**:3789–3801.

[18] Zou KH, O'Malley AJ, Mauri L. Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models. *Circulation*. 2007; **115**:654–657.

[19] Mazurenko, S., Prokop, Z., and Damborsky, J. (2019). Machine learning in enzyme engineering. ACS Catalysis, 10(2), 1210-1223.

[20] Yang, Y.; Niroula, A.; Shen, B.; Vihinen, M. PON-Sol: Prediction of Effects of Amino Acid Substitutions on Protein Solubility. Bioinformatics 2016, 32, 2032!2034.

[21] Folkman, L.; Stantic, B.; Sattar, A.; Zhou, Y. EASE-MM: Sequence-Based Prediction of Mutation-Induced Stability Changes with Feature-Based Multiple Models. J. Mol. Biol. 2016, 428, 1394! 1405.

[22] Teng, S.; Srivastava, A. K.; Wang, L. Sequence Feature-Based Prediction of Protein Stability Changes upon Amino Acid Substitutions. BMC Genomics 2010, 11, S5.

[23] Huang, L.; Gromiha, M. M.; Ho, S. iPTREE-STAB: Interpretable Decision Tree Based Method for Predicting Protein Stability Changes Upon Mutations. Bioinformatics 2007, 23, 1292! 1293.

[24] Koskinen, P.; Toronen, P.; Nokso-Koivisto, J.; Holm, L. PANNZER: High-Throughput Functional Annotation of Uncharac- terized Proteins in an Error-Prone Environment. Bioinformatics 2015, 31, 1544!1552.

[25] De Ferrari, L.; Mitchell, J. B. From Sequence to Enzyme Mechanism Using Multi-Label Machine Learning. BMC Bioinf. 2014, 15, 150.

[26] Falda, M.; Toppo, S.; Pescarolo, A.; Lavezzo, E.; Di Camillo, B.; Facchinetti, A.; Cilia, E.; Velasco, R.; Fontana, P. Argot2: A Large Scale Function Prediction Tool Relying on Semantic Similarity of Weighted Gene Ontology Terms. BMC Bioinf. 2012, 13, S14.

[27] Cozzetto, D.; Buchan, D. W.; Bryson, K.; Jones, D. T. Protein Function Prediction by Massive Integration of Evolutionary Analyses and Multiple Data Sources. BMC Bioinf. 2013, 14, S1.

[28] Kulski, J. Next Generation Sequencing: Advances, Applications and Challenges; InTechOpen: London, 2016.

[29] Straiton, J.; Free, T.; Sawyer, A.; Martin, J. From Sanger Sequencing to Genome Databases and Beyond. BioTechniques 2019, 66, 60-63.

[30] Ardui, S.; Ameur, A.; Vermeesch, J. R.; Hestand, M. S. Single Molecule Real-Time (SMRT) Sequencing Comes of Age: Applications and Utilities for Medical Diagnostics. Nucleic Acids Res. 2018, 46, 2159-2168.

[31] Kono, N., and Arakawa, K. (2019). Nanopore sequencing: Review of potential applications in functional genomics. Development, growth and differentiation, 61(5), 316-326.

[32] Bunzel, H. A., Garrabou, X., Pott, M., and Hilvert, D. (2018). Speeding up enzyme discovery and engineering with ultrahigh-throughput methods. Current opinion in structural biology, 48, 149-156.

[33] Wrenbeck, E. E., Faber, M. S., and Whitehead, T. A. (2017). Deep sequencing methods for protein engineering and design. Current opinion in structural biology, 45, 36-44.

[34] Fowler, D. M., and Fields, S. (2014). Deep mutational scanning: a new style of protein science. Nature methods, 11(8), 801-807.

[35] Gupta, K., and Varadarajan, R. (2018). Insights into protein structure, stability and function from saturation mutagenesis. Current opinion in structural biology, 50, 117-125.

[36] UniProt Consortium. UniProt: A Worldwide Hub of Protein Knowledge. Nucleic Acids Res. 2018, 47, D506-D515.

[37] Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Zidek, A.; Nelson, A.; Bridgland, A.; Penedones, H.; Petersen, S.; Simonyan, K.; Jones, D. T.; Silver, D.; Kavukcuoglu, K.; Hassabis, D.; Senior, A. W. De Novo Structure Prediction with Deeplearning Based Scoring. In Thirteenth Critical Assessment of Techniques for Protein Structure Prediction Abstracts; 2018; pp 11-12.

[38] Kinch, L. N.; Shi, S.; Cheng, H.; Cong, Q.; Pei, J.; Mariani, V.; Schwede, T.; Grishin, N. V. CASP9 Target Classification. Proteins: Struct., Funct., Genet. 2011, 79, 21-36.

[39] Shehu, A.; Barbará, D.; Molloy, K. A Survey of ComputationalMethods for Protein Function Prediction. In Big Data Analytics in Genomics; Wong, K. C., Ed.; Springer: Cham, 2016; pp 225-298.

[40] Zhang, C.; Freddolino, P. L.; Zhang, Y. COFACTOR: Improved Protein Function Prediction by Combining Structure, Sequence and Protein-Protein Interaction Information. Nucleic Acids Res. 2017, 45, W291-W299.

[41] Kumar, N.; Skolnick, J. EFICAz2. 5: Application of a High-Precision Enzyme Function Predictor to 396 Proteomes. Bioinformatics 2012, 28, 2687-2688.

[42] Li, Y.; Wang, S.; Umarov, R.; Xie, B.; Fan, M.; Li, L.; Gao, X. DEEPre: Sequence-Based Enzyme EC Number Prediction by Deep Learning. Bioinformatics 2018, 34, 760-769.

[43] Yang, M.; Fehl, C.; Lees, K. V.; Lim, E. K.; Offen, W. A.; Davies, G. J.; Bowles, D. J.; Davidson, M. G.; Roberts, S. J.; Davis, B. G. Functional and Informatics Analysis Enables Glycosyltransferase Activity Prediction. Nat. Chem. Biol. 2018, 14, 1109-1117.

[44] Niwa, T.; Ying, B. W.; Saito, K.; Jin, W.; Takada, S.; Ueda, T.; Taguchi, H. Bimodal Protein Solubility Distribution Revealed by an Aggregation Analysis of the Entire Ensemble of Escherichia Coli Proteins. Proc. Natl. Acad. Sci. U. S. A. 2009, 106, 4201-4206.

[45] Klesmith, J. R.; Bacik, J. P.; Wrenbeck, E. E.; Michalczyk, R.; Whitehead, T. A. Trade-Offs Between Enzyme Fitness and Solubility Illuminated by Deep Mutational Scanning. Proc. Natl. Acad. Sci. U. S. A. 2017, 114, 2265-2270.

[46] Ruiz-Blanco, Y. B.; Paz, W.; Green, J.; Marrero-Ponce, Y. ProtDCal: A Program to Compute General-Purpose-Numerical Descriptors for Sequences and 3D-Structures of Proteins. BMC Bioinf. 2015, 16, 162.

[47] Han, X.; Wang, X.; Zhou, K. Develop Machine Learning-Based Regression Predictive Models for Engineering Protein Solubility. Bioinformatics 2019, 35, 4640-4646.

[48] Musil, M.; Konegger, H.; Hon, J.; Bednar, D.; Damborsky, J. Computational Design of Stable and Soluble Biocatalysts. ACS Catal. 2019, 9, 1033-1054.

[49] Li, G.; Dong, Y.; Reetz, M. T. Can Machine Learning Revolutionize Directed Evolution of Selective Enzymes? Adv. Synth. Catal. 2019, 361, 2377-2386.

[50] Wu, Z.; Kan, S. B. J.; Lewis, R. D.; Wittmann, B. J.; Arnold, F. H. Machine Learning-Assisted Directed Protein Evolution with Combinatorial Libraries. Proc. Natl. Acad. Sci. U. S. A. 2019, 116, 8852-8858.

[51] Wolpert, D. H.; Macready, W. G. No Free Lunch Theorems for Optimization. IEEE Trans. Evol. Comput. 1997, 1, 67-82.

[52] Wolpert, D. H. The Lack of a Priori Distinctions between Learning Algorithms. Neural Comput. 1996, 8, 1341-1390.

[53] Walsh, I.; Pollastri, G.; Tosatto, S. C. Correct Machine Learning on Protein Sequences: A Peer-Reviewing Perspective. Briefings Bioinf. 2016, 17, 831-840.

[54] Rao, R.; Bhattacharya, N.; Thomas, N.; Duan, Y.; Chen, X.; Canny, J.; Abbeel, P.; Song, Y. S. Evaluating Protein Transfer Learning with TAPE. arXiv preprint arXiv:1906.08230, 2019.

[55] Romero, P. A.; Krause, A.; Arnold, F. H. Navigating the Protein Fitness Landscape with Gaussian Processes. Proc. Natl. Acad. Sci. U. S. A. 2013, 110, E193-E201

[56] Eraslan, G.; Avsec, Z; Gagneur, J.; Theis, F. J. Deep Learning: New Computational Modelling Techniques for Genomics. Nat. Rev. Genet. 2019, 20, 389-403.

[57] Repecka, D.; Jauniskis, V.; Karpus, L.; Rembeza, E.; Zrimec, J.; Poviloniene, S.; Rokaitis, I.; Laurynenas, A.; Abuajwa, W.; Savolainen, O.; Meskys, R.; Engqvist, M. K. M.; Zelezniak, A. Expanding Functional Protein Sequence Space Using Generative Adversarial Networks. bioRxiv 2019, DOI: 10.1101/789719.

[58] Riesselman, A. J.; Ingraham, J. B.; Marks, D. S. Deep Generative Models of Genetic Variation Capture the Effects of Mutations. Nat. Methods 2018, 15, 816-822.

[59] Thornton, C.; Hutter, F.; Hoos, H. H.; Leyton-Brown, K. Auto- WEKA: Combined Selection and Hyperparameter Optimization of Classiffication Algorithms. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining; 2013; pp 847-855.

[60] Polikar, R. Ensemble based systems in decision making. IEEE Circuits and systems magazine 2006, 6, 21-45.

[61] Gammerman, A.; Vovk, V. Hedging Predictions in Machine Learning. Comput. J. 2007, 50, 151-163.

[62] Samek, W.; Wiegand, T.; Müller, K. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. ITU Journal: ICT Discoveries 2017, 39-48.

[63] Shrikumar, A.; Greenside, P.; Kundaje, A. Learning Important Features through Propagating Activation differences. In Proceedings of the 34th International Conference on Machine Learning; 2017; Vol. 70, pp 3145-3153.

[64] Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classiffication Models and Saliency Maps. arXiv preprint arXiv:1312.6034 2013.

[65] Brookes, D. H.; Park, H.; Listgarten, J. Conditioning by Adaptive Sampling for Robust Design. In Proceedings of the 36th International Conference on Machine Learning; 2019; Vol. 97, pp 773-782.

[66] Ribeiro, M. T.; Singh, S.; Guestrin, C. "Why Should I Trust You?" Explaining the Predictions of Any Classiffier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016; pp 1135-1144.

[67] Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing Properties of Neural Networks. arXiv preprint arXiv:1312.6199 201

[68] Yu, M. K.; Ma, J.; Fisher, J.; Kreisberg, J. F.; Raphael, B. J.; Ideker, T. Visible Machine Learning for Biomedicine. Cell 2018, 173, 1562-1565.

[69] Copley, S. D. Shining a light on enzyme promiscuity. Curr. Opin. Struct. Biol. 47, 167–175 (2017).

[70] Nobeli, I., Favia, A. D. and Thornton, J. M. Protein promiscuity and its implications for biotechnology. Nat. Biotechnol. 27, 157–167 (2009)

[71] Adrio, J. L. and Demain, A. L. Microbial enzymes: tools for biotechnological processes. Biomolecules 4, 117–139 (2014).

[72] Wang, S. et al. Engineering a synthetic pathway for gentisate in pseudomonas chlororaphis p3. Front. Bioeng. Biotechnol. 8, 1588 (2021).

[73] Wu, M.-C., Law, B., Wilkinson, B. and micklefied, J. Bioengineering natural product biosynthetic pathways for therapeutic applications. Curr. Opin. Biotechnol. 23, 931–940 (2012)

[74] Rembeza, E., Boverio, A., Fraaije, M. W. and Engqvist, M. K. Discovery of two novel oxidases using a high-throughput activity screen. ChemBioChem 23, e202100510 (2022).

[75] Longwell, C. K., Labanieh, L. and Cochran, J. R. High-throughput screening technologies for enzyme engineering. Curr. Opin. Biotechnol. 48, 196–202 (2017).

[76] Black, G. W. et al. A high-throughput screening method for determining the substrate scope of nitrilases. Chem. Commun. 51, 2660–2662 (2015).

[77] Pertusi, D. A. et al. Predicting novel substrates for enzymes with minimal experimental effort with active learning. Metab. Eng. 44,171-181 (2017).

[78] Mou, Z. et al. Machine learning-based prediction of enzyme substrate scope: Application to bacterial nitrilases. Proteins Struct. Funct. Bioinf. 89, 336-347 (2021).

[79] Yang, M. et al. Functional and informatics analysis enables glycosyltransferase activity prediction. Nat. Chem. Biol. 14, 1109–1117 (2018).

[80] Rottig, M., Rausch, C. and Kohlbacher, O. Combining structure and sequence information allows automated prediction of substrate specificities within enzyme families. PLoS Comput. Biol. 6, e1000636 (2010).

[81] Chevrette, M. G., Aicheler, F., Kohlbacher, O., Currie, C. R. and Medema, M. H. Sandpuma: ensemble predictions of nonribosomal peptide chemistry reveal biosynthetic diversity across actinobacteria. Bioinformatics 33, 3202-3210 (2017).

[82] Goldman, S., Das, R., Yang, K. K. and Coley, C. W. Machine learning modeling of family wide enzyme-substrate specificity screens. PLoS Comput. Biol. 18, e1009853 (2022).

[83] Visani, G. M., Hughes, M. C. and Hassoun, S. Enzyme promiscuity prediction using hierarchy-informed multi-label classiffication Bioinformatics 37, 2017-2024 (2021).

[84] Ryu, J. Y., Kim, H. U. and Lee, S. Y. Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. PNAS 116, 13996-14001 (2019).

[85] Li, Y. et al. DEEPre: sequence-based enzyme EC number prediction by deep learning. Bioinformatics 34, 760-769 (2017).

[86] Sanderson, T., Bileschi, M. L., Belanger, D. and Colwell, L. J. Proteinfer, deep neural networks for protein functional inference. eLife 12, e80942 (2023).

[87] Bileschi, M. L. et al. Using deep learning to annotate the protein universe. Nat. Biotechnol.https://doi.org/10.1038/s41587-021-01179-w (2022).

[88] Rembeza, E. and Engqvist, M. K. Experimental and computational investigation of enzyme functional annotations uncovers misannotation in the ec 1.1. 3.15 enzyme class. PLoS Comput. Biol. 17, e1009446 (2021).

[89] Ozturk, H., Ozgur, A. and Ozkirimli, E. Deepdta: deep drugtarget binding affinity prediction. Bioinformatics 34, i821-i829 (2018).

[90] Feng, Q., Dueva, E., Cherkasov, A. and Ester, M. Padme: A deep learning-based framework for drug-target interaction prediction. Preprint at https://doi.org/10.48550/arXiv.1807.09741 (2018).

[91] Karimi, M., Wu, D., Wang, Z. and Shen, Y. Deep affinity: interpretable deep learning of compound–protein affinity through UNIFIED recurrent and convolutional neural networks. Bioinformatics 35, 3329-3338 (2019).

[92] Kroll, A., Engqvist, M. K., Heckmann, D. and Lercher, M. J. Deep learning allows genome-scale prediction of michaelis constants from structural features. PLoS Biol. 19, e3001402 (2021).

[93] Li, F. et al. Deep learning-based k cat prediction enables improved enzyme-constrained model reconstruction. Nat. Catal. 5, 662-672 (2022).

[94] Weininger, D. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. J. Chem. Inf. Comput. Sci. 28, 31-36 (1988).

[95] Rogers, D. and Hahn, M. Extended-connectivity fingerprints. J. Chem. Inf. Model. 50, 742-754 (2010).

[96] Zhou, J. et al. Graph neural networks: A review of methods and applications. AI Open 1, 57-81 (2020).

[97] Yang, K. et al. Analyzing learned molecular representations for property prediction. J. Chem. Inf. Model. 59, 3370-3388 (2019).

[98] Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. PNAS 118, e2016239118 (2021).

[99] Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. and Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. Nat. Methods. 16, 1315-1322 (2019).

[100] Xu, Y. et al. Deep dive into machine learning models for protein engineering. J. Chem. Inf. Model. 60, 2773–2790 (2020).

[101] Bekker, J. and Davis, J. Learning from positive and unlabeled data: A survey. Mach. Learn. 109, 719-760 (2020)

[102] Kearnes, S., McCloskey, K., Berndl, M., Pande, V. and Riley, P. Mole- cular graph convolutions: moving beyond !ngerprints. J. Comput. -Aided Mol. Des. 30, 595–608 (2016).

[103] Duvenaud, D. K. et al. Convolutional networks on graphs for learning molecular fingerprints. In Advances in Neural Information Processing Systems, 2224-2232 (2015).

[104] Zhou, J. et al. Graph neural networks: A review of methods and applications. AI Open 1, 57–81 (2020).

[105] Hu, W. et al. Strategies for pre-training graph neural networks. Preprint at https://doi.org/10.48550/arXiv.1905.12265 (2019).

[106] Capela, F., Nouchi, V., Van Deursen, R., Tetko, I. V. and Godin, G. Multitask learning on graph neural networks applied to molecular property predictions. Preprint at https://doi.org/10.48550/arXiv.1910.13124 (2019).

[107] Vaswani, A. et al. Attention is all you need. In Advances in neural information processing systems, 5998–6008 (2017).

[108] Suzek, B. E. et al. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinfor- matics 31, 926–932 (2015).

[109] Elnaggar, A. et al. Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. IEEE Trans. Pattern Anal. Mach. Intell. PP https://doi.org/10.1109/TPAMI.2021.3095381 (2021).

[110] Wittmann, B. J., Johnston, K. E., Wu, Z., and Arnold, F. H. (2021). Advances in machine learning for directed evolution. Current opinion in structural biology, 69, 11-18.