Lectura 5: Conceptos y modelos fundamentales (regresión lineal, regresión logística)

SI3015 - Fundamentos de Aprendizaje Automático

2026

1. Regresión Lineal

La regresión lineal constituye uno de los pilares fundamentales del aprendizaje supervisado, actuando como un modelo paramétrico diseñado para modelar la relación funcional entre una variable dependiente continua y una o más variables independientes.

1.1. Regresión Lineal Simple

En un escenario de **2D** (dos dimensiones), la regresión lineal es la forma más pura y visual del algoritmo, conocida técnicamente como **Regresión Lineal Simple**. En este contexto, se trabaja con una sola variable predictora (x) y una variable de respuesta (y).

1.1.1. La Geometría del Modelo

El objetivo es encontrar la **línea recta** que mejor represente la tendencia de un conjunto de puntos dispersos en un plano cartesiano. Matemáticamente, el modelo se define por la ecuación de la recta:

$$y = \beta_0 + \beta_1 x \tag{1}$$

- β_1 **(Pendiente):** Determina la inclinación de la recta. Indica cuánto cambia y por cada unidad que aumenta x.
- β_0 **(Intercepto):** Es el punto donde la recta cruza el eje vertical (y), es decir, el valor de y cuando $x = 0$.

1.1.2. El Ajuste: Mínimos Cuadrados Ordinarios (OLS)

Para que la línea sea "la mejor", aplicamos el criterio de **Mínimos Cuadrados**. Se mide la distancia vertical desde cada punto dato real hasta la línea de predicción. Esa distancia se denomina **residuo**.

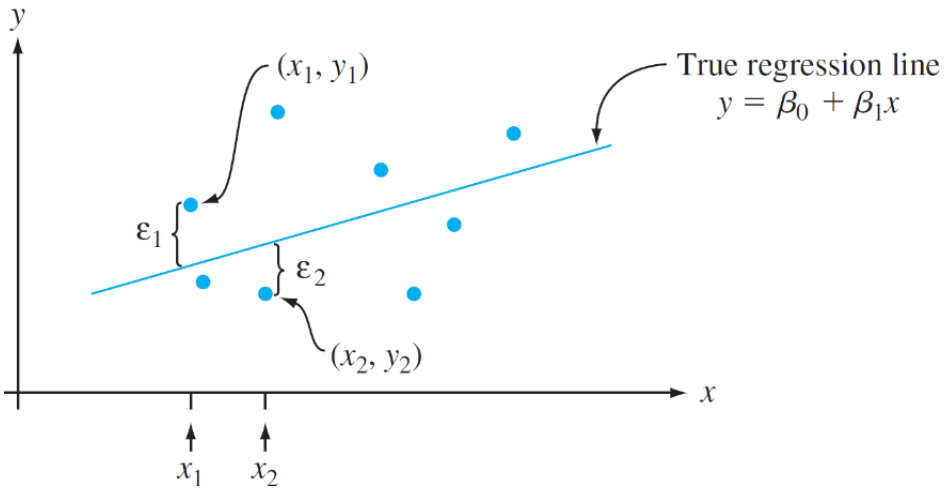

Figura 1: Regresión lineal simple

El algoritmo busca los valores de β_0 y β_1 que minimicen la suma de esos residuos (o errores) elevados al cuadrado:

$$\text{Minimizar} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2}$$

Se eleva al cuadrado para evitar que los errores negativos se cancelen con los positivos y para penalizar de forma cuadrática las desviaciones mayores. Para aplicar este método se deben aplicar los siguientes pasos:

Función de Pérdida: Suma de Cuadrados de los Residuos
Para cada observación (x_i, y_i), definimos el residuo e_i como la diferencia entre el valor real y el predicho por el modelo:

$$e_i = y_i - \hat{y}_i = y_i - (\beta_0 + \beta_1 x_i) \tag{3}$$

El objetivo de MCO es minimizar la función de costo $J(\beta_0, \beta_1)$, también conocida como la Suma de los Cuadrados de los Residuos (RSS):

$$J(\beta_0, \beta_1) = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \tag{4}$$

Derivación de las Fórmulas Cerradas Para minimizar J, se calculan las derivadas parciales respecto a cada parámetro y se igualan a cero. Esto nos da las fórmulas cerradas para calcular los coeficientes directamente desde los datos:

A.Cálculo de la Pendiente (β_1) La pendiente óptima se obtiene mediante la relación entre la covarianza muestral de (x, y) y la varianza muestral de x:

$$\beta_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{5}$$

B.Cálculo del Intercepto (β_0) Una vez determinado β_1, el intercepto se calcula asegurando que la recta pase por el centroide de los datos $(\bar{x}, \bar{y})$:

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \tag{6}$$

Donde $\bar{x}$ y $\bar{y}$ representan las medias aritméticas de las variables independientes y dependientes, respectivamente.

Procedimiento Paso a Paso:

Si se requiere aplicar el método de Mínimos Cuadrados Ordinarios de forma manual, el flujo algorítmico es el siguiente:

1. **Calcular las medias aritméticas:** Se determinan los valores promedio de las observaciones para ambas variables:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{7}$$

2. **Calcular las desviaciones:** Para cada punto i, se obtienen las diferencias respecto a la media: $(x_i - \bar{x})$ y $(y_i - \bar{y})$.

3. **Calcular el coeficiente de pendiente (β_1):** Se computa como el cociente entre la suma de los productos de las desviaciones y la suma de los cuadrados de las desviaciones de x:

$$\beta_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \tag{8}$$

4. **Calcular el intercepto (β_0):** Se despeja utilizando los valores obtenidos previamente y las medias:

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \tag{9}$$

5. **Construir la ecuación final:** Se define la función de predicción del modelo:

$$\hat{y} = \beta_0 + \beta_1 x \tag{10}$$

Propiedades Técnicas

- **Eliminación de signos:** El uso de potencias cuadráticas evita que los errores positivos y negativos se compensen entre sí.
- **Penalización de Outliers:** Debido a la naturaleza cuadrática, las desviaciones grandes tienen un impacto desproporcionado en la función de costo, obligando al modelo a ajustarse más cerca de los valores atípicos.

1.1.3. Interpretación Visual

- **Correlación Positiva:** Si la recta tiene pendiente positiva ($\beta_1 > 0$), las variables crecen proporcionalmente.
- **Correlación Negativa:** Si la recta tiene pendiente negativa ($\beta_1 < 0$), una variable crece mientras la otra decrece.
- **Sin Correlación:** Si la recta es aproximadamente horizontal ($\beta_1 \approx 0$), la variable x no posee poder predictivo sobre y.

1.1.4. Componentes Clave

Concepto	**Representación Física**	**Función**
Variable Independiente (x)	Eje Horizontal (Abscisas)	El predictor o causa.
Variable Dependiente (y)	Eje Vertical (Ordenadas)	El target o efecto.
Residual (e)	Distancia vertical punto-línea	El error de predicción.

1.1.5. Ejemplo

Se desea modelar la relación entre la superficie de una vivienda (x, en metros cuadrados) y su precio de venta (y, en miles de USD). Para ello, se dispone de un conjunto de datos histórico de cinco observaciones.

La siguiente tabla muestra la correspondencia entre los metros cuadrados y el precio observado:

Metros Cuadrados (x)	**Precio en miles de USD (y)**
50	150
70	200
85	240
100	290
120	350

Tras aplicar el método de **Mínimos Cuadrados Ordinarios**, se obtiene la siguiente ecuación de la recta de mejor ajuste:

$$\hat{y} = 2{,}8x + 2 \tag{11}$$

Para llegar a la anterior ecuación y dados los datos de la siguiente tabla, procedemos al cálculo de los parámetros del modelo $\hat{y} = \beta_0 + \beta_1 x$:

1. Cálculo de Medias

$$\bar{x} = \frac{425}{5} = 85, \quad \bar{y} = \frac{1230}{5} = 246 \tag{12}$$

x_i	y_i	$(x_i - \bar{x})$	$(y_i - \bar{y})$	$(x_i - \bar{x})(y_i - \bar{y})$	$(x_i - \bar{x})^2$
50	150	-35	-96	3360	1225
70	200	-15	-46	690	225
85	240	0	-6	0	0
100	290	15	44	660	225
120	350	35	104	3640	1225
Total				**8350**	**2900**

Cuadro 1: Desviaciones y productos cruzados para el cálculo de β_1.

2. Cálculo de la Pendiente (β_1) Aplicando la sumatoria de productos de las desviaciones:

$$\beta_1 = \frac{\sum_{i=1}^{5}(x_i - 85)(y_i - 246)}{\sum_{i=1}^{5}(x_i - 85)^2} = \frac{8350}{2900} \approx 2{,}8 \tag{13}$$

3. Cálculo del Intercepto (β_0)

$$\beta_0 = 246 - (2{,}8 \times 85) \approx 2 \tag{14}$$

Donde los coeficientes se interpretan de la siguiente manera:

- **Pendiente ($\beta_1 = 2{,}8$):** Representa el costo marginal por metro cuadrado. Por cada unidad incremental de x, el precio aumenta en \$2,800.
- **Intercepto ($\beta_0 = 2$):** Representa el valor base del inmueble (por ejemplo, el valor del terreno) cuando la construcción es nula.

Para estimar el valor de una propiedad de $90\,m^2$, se realiza la sustitución en el modelo:

$$\hat{y} = 2{,}8(90) + 2 = 252 + 2 = 254 \tag{15}$$

El valor predicho para dicha propiedad es de **\$254,000 USD**.

El error de predicción o residuo (e) se define como la diferencia entre el valor real y el valor estimado:

$$e_i = y_i - \hat{y}_i \tag{16}$$

Para la observación de $100\,m^2$ (donde el precio real es 290):

$$e = 290 - (2{,}8(100) + 2) = 290 - 282 = 8 \tag{17}$$

El modelo subestima el valor real por \$8,000 USD en este punto específico.

1.2. Regresión Lineal Múltiple

La **Regresión Lineal** es un modelo paramétrico de aprendizaje supervisado que asume una relación lineal entre un conjunto de variables explicativas (features) y una variable de respuesta continua (target).

El objetivo es estimar una función de hipótesis $h_\theta(x)$ que aproxime el valor real y. Para un escenario de regresión lineal múltiple con n características, el modelo se define como:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \tag{18}$$

En notación vectorial (representación compacta para computación), esto se expresa como el producto punto entre el vector de pesos y el vector de características:

$$\hat{y} = h_\theta(x) = \theta^T \mathbf{x} \tag{19}$$

Donde:

- $\mathbf{x}$: Es el vector de entrada (incluyendo un término de sesgo $x_0 = 1$).
- θ: Es el vector de parámetros (pesos) que el modelo debe aprender.
- $\hat{y}$: Es el valor escalar predicho.

1.3. Función de Costo (Loss Function)

Para medir la precisión del ajuste, utilizamos generalmente el **Error Cuadrático Medio (MSE)**. Buscamos minimizar la suma de los residuos al cuadrado respecto a los parámetros θ:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{20}$$

Donde m es el número de ejemplos en el conjunto de datos.

1.4. Optimización

Existen dos vías principales para hallar el vector de parámetros óptimo θ:

1.4.1. Ecuación Normal

Es una solución analítica directa basada en el cálculo matricial. Se deriva igualando el gradiente de la función de costo a cero:

$$\theta = (X^T X)^{-1} X^T y \tag{21}$$

Nota: Presenta una complejidad computacional de $O(n^3)$, lo que la hace costosa para grandes volúmenes de datos.

1.4.2. Descenso de Gradiente (Gradient Descent)

Es un algoritmo iterativo que actualiza los pesos en la dirección opuesta al gradiente de la función de costo:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \tag{22}$$

Donde α es la **tasa de aprendizaje** (learning rate).

1.5. Supuestos Estadísticos (Asunciones de Gauss-Markov)

Para que el modelo sea el **Mejor Estimador Lineal Insesgado (BLUE)**, debe cumplir:

1. **Linealidad:** La relación entre las variables es lineal en los parámetros.
2. **Homocedasticidad:** $Var(\epsilon|X) = \sigma^2 I$ (varianza de errores constante).
3. **Independencia:** No existe autocorrelación entre los residuos.
4. **Normalidad:** Los errores siguen una distribución $N(0, \sigma^2)$.
5. **No multicolinealidad:** Las variables independientes no están altamente correlacionadas.

1.6. Regularización

La **regularización** es una técnica esencial en el aprendizaje automático para controlar la complejidad de un modelo. Cuando un modelo tiene demasiados parámetros o estos son excesivamente grandes, tiende a memorizar el ruido de los datos de entrenamiento en lugar de aprender el patrón general, un fenómeno conocido como **sobreajuste (overfitting)**.

Para solucionar esto, modificamos la función de costo original $J(\theta)$ añadiendo un término de penalización que castiga los coeficientes θ elevados.

- **L1 (Lasso):** Añade el término $\lambda \sum |\theta_j|$.
- **L2 (Ridge):** Añade el término $\lambda \sum \theta_j^2$.

Un modelo sobreajustado posee una varianza alta: funciona de manera óptima con los datos de entrenamiento, pero falla significativamente al enfrentarse a datos nuevos (datos de prueba). La regularización introduce un ligero **sesgo** en el entrenamiento a cambio de una reducción drástica en la **varianza**, mejorando la capacidad de generalización del algoritmo.

1.6.1. Regresión Ridge (Regularización L2)

La regresión Ridge añade el cuadrado de la magnitud de los coeficientes a la función de costo original. Matemáticamente se expresa como:

$$J(\theta)_{Ridge} = \text{MSE} + \lambda \sum_{j=1}^{n} \theta_j^2 \tag{23}$$

- **Efecto:** Reduce proporcionalmente todos los pesos θ, pero **nunca los hace exactamente cero**.
- **Uso ideal:** Es recomendable cuando se dispone de muchas variables que contribuyen de manera pequeña pero constante al resultado. Es particularmente útil para manejar la **multicolinealidad** (correlación alta entre variables independientes).

1.6.2. Regresión Lasso (Regularización L1)

La técnica Lasso (*Least Absolute Shrinkage and Selection Operator*) añade el valor absoluto de la magnitud de los coeficientes:

$$J(\theta)_{Lasso} = \text{MSE} + \lambda \sum_{j=1}^{n} |\theta_j| \tag{24}$$

- **Efecto:** Debido a la geometría de la penalización L1, Lasso tiene la propiedad de forzar que los coeficientes de las variables menos relevantes sean **exactamente cero**.
- **Uso ideal:** Funciona como un método intrínseco de **selección de características** (*feature selection*). Es sumamente útil en conjuntos de datos con una gran cantidad de variables donde se sospecha que solo un subconjunto de ellas es realmente significativo.

1.6.3. El Hiperparámetro λ (Lambda)

En ambas técnicas, el parámetro λ controla la intensidad de la regularización:

- Si $\lambda = 0$: El modelo se comporta como una regresión lineal estándar por Mínimos Cuadrados Ordinarios.
- Si $\lambda \to \infty$: Los pesos se reducen tanto que el modelo se vuelve demasiado simple, incurriendo en **subajuste (underfitting)**.

2. Regresión Logística

La **regresión logística** es un modelo estadístico y de aprendizaje automático supervisado utilizado para predecir la probabilidad de una variable dependiente categórica. A diferencia de la regresión lineal, se utiliza para problemas de **clasificación** (generalmente binaria).

2.1. La Función Sigmoide (Logística)

Dado que la salida de una combinación lineal $\theta^T \mathbf{x}$ puede variar en el rango $(-\infty, +\infty)$, la regresión logística utiliza una función de activación denominada **sigmoide** para mapear cualquier número real al intervalo $(0, 1)$:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{25}$$

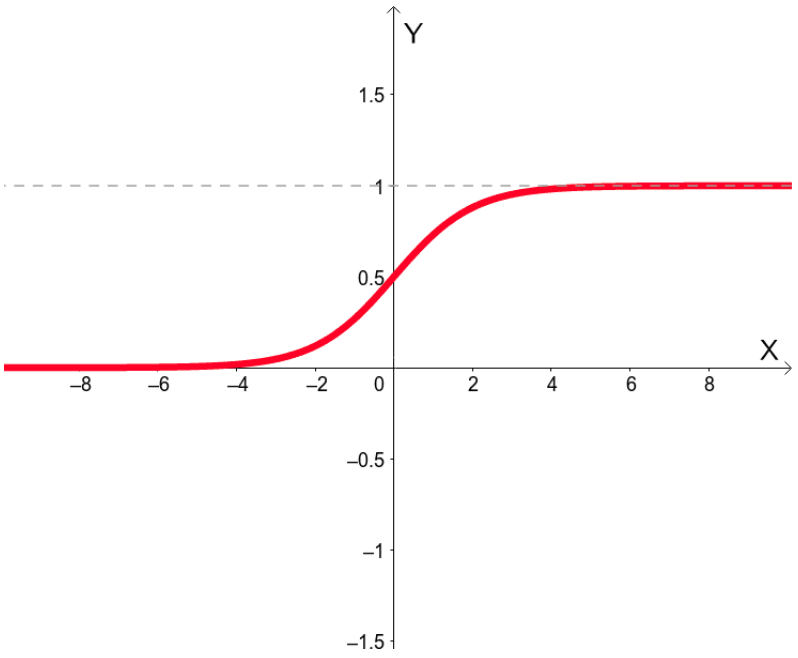

Figura 2: Función Sigmoide

2.2. Comportamiento Asintótico y Rango

La función sigmoide, también conocida como curva logística, actúa como un mecanismo de *squashing* (aplastamiento). Independientemente de la magnitud del valor de entrada z (donde $z = \theta^T \mathbf{x}$ en el contexto de regresión), el resultado se encuentra estrictamente acotado en el intervalo abierto $(0, 1)$.

Sus límites fundamentales son:

- **Límite Superior:** A medida que $z \to \infty$, el término $e^{-z} \to 0$, por lo cual $\sigma(z) \to 1$.

- **Límite Inferior:** A medida que $z \to -\infty$, el término $e^{-z} \to \infty$, lo que implica que $\sigma(z) \to 0$.
- **Punto de Inflexión:** Cuando $z = 0$, $\sigma(0) = 0{,}5$. Este valor se utiliza habitualmente como el umbral de decisión (*threshold*) estándar para la clasificación binaria.

2.3. Propiedad de la Derivada y Eficiencia Computacional

Una de las ventajas críticas de la función sigmoide en el aprendizaje automático es la elegancia de su derivada. Para algoritmos de optimización como el Descenso de Gradiente, es necesario calcular gradientes de forma iterativa. La derivada de $\sigma(z)$ puede expresarse en términos de la propia función:

$$\frac{d}{dz}\sigma(z) = \sigma(z)(1 - \sigma(z)) \tag{26}$$

Esta propiedad es extremadamente valiosa desde el punto de vista computacional, ya que permite reutilizar el valor calculado durante la propagación hacia adelante (*forward pass*) para obtener el gradiente durante la retropropagación (*backpropagation*), reduciendo la carga de cálculo.

2.4. Relación con el Logit (Log-Odds)

La función sigmoide es matemáticamente la inversa de la función **logit**. Si definimos $p = \sigma(z)$ como la probabilidad de que ocurra el evento de interés, podemos derivar la relación inversa:

$$z = \ln\left(\frac{p}{1-p}\right) \tag{27}$$

En esta expresión, la razón $\frac{p}{1-p}$ se denomina *odds ratio* (razón de probabilidades). Al aplicar el logaritmo natural, obtenemos el **logit**. Por lo tanto, en la regresión logística, lo que realmente se está modelando es el logaritmo de las probabilidades como una combinación lineal de las variables de entrada:

$$\ln\left(\frac{P(y=1|x)}{1-P(y=1|x)}\right) = \theta^T \mathbf{x} \tag{28}$$

2.5. Función de Hipótesis

La hipótesis representa la probabilidad condicional de que un ejemplo pertenezca a la clase positiva ($y = 1$) dada la entrada $\mathbf{x}$ y los parámetros θ:

$$h_\theta(x) = P(y=1|x;\theta) = \sigma(\theta^T \mathbf{x}) = \frac{1}{1+e^{-\theta^T \mathbf{x}}} \tag{29}$$

La decisión de clasificación se basa típicamente en un umbral crítico:

- Si $h_\theta(x) \geq 0{,}5 \Rightarrow \hat{y} = 1$
- Si $h_\theta(x) < 0{,}5 \Rightarrow \hat{y} = 0$

2.6. Función de Costo: Log Loss (Entropía Cruzada)

Para la optimización, se utiliza la **Entropía Cruzada Binaria**, ya que produce una superficie de error convexa. La función de costo $J(\theta)$ para m ejemplos se define como:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] \tag{30}$$

Esta función penaliza logarítmicamente las predicciones incorrectas realizadas con alta confianza.

2.7. Optimización

El vector de parámetros θ se actualiza iterativamente mediante el **Descenso de Gradiente**:

$$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \tag{31}$$

2.8. Comparativa Técnica

Característica	**Regresión Lineal**	**Regresión Logística**
Variable de salida	Continua	Categórica (Probabilidad)
Ecuación de salida	$\theta^T\mathbf{x}$	$\sigma(\theta^T\mathbf{x})$
Función de Costo	Error Cuadrático Medio	Entropía Cruzada
Uso principal	Regresión (Predicción)	Clasificación

2.9. Ejemplo

En este caso de estudio, se analiza la bandeja de entrada de un servidor de correo para predecir si un mensaje es **SPAM (1)** o **Correo Seguro (0)**. La predicción se basa en una única variable independiente: la cantidad de palabras sospechosas (como "gratis", "oferta" o "ganaste") identificadas en el cuerpo del mensaje.

El modelo se entrena utilizando los siguientes datos históricos recolectados:

Correo	**Palabras Sospechosas (x)**	**Resultado Real (y)**
A	1	0 (Seguro)
B	2	0 (Seguro)
C	4	1 (Spam)
D	6	1 (Spam)
E	9	1 (Spam)

Cuadro 2: Dataset para el entrenamiento del filtro de correo.

2.9.1. Aplicación y Mecanismo del Modelo

A diferencia de la regresión lineal, la logística no predice un conteo, sino la probabilidad de pertenencia a una clase. Si tras el entrenamiento el modelo determina ciertos pesos (parámetros), el proceso para un nuevo correo con $x = 3$ palabras sospechosas es el siguiente:

1. **Cálculo del Valor Lineal (z):** Se aplica la combinación lineal de la entrada. Supongamos que el modelo determinó la función $z = 1{,}5x - 4{,}5$. Para $x = 3$:
$$z = 1{,}5(3) - 4{,}5 = 0 \tag{32}$$

2. **Transformación Sigmoide:** El valor z se introduce en la función de activación para obtener un rango entre 0 y 1:
$$P(y = 1) = \frac{1}{1 + e^{-0}} = 0{,}5 \tag{33}$$
Esto indica una probabilidad del 50 % de que el correo sea Spam.

3. **Umbral de Decisión (Threshold):** El sistema utiliza un punto de corte para la clasificación final. Si el correo tuviera 5 palabras sospechosas, la probabilidad ascendería aproximadamente al 95 %, superando el umbral estándar de 0.5 y marcándose automáticamente como **SPAM**.

2.9.2. Resumen del Proceso

El flujo lógico del algoritmo consiste en: tomar las características de entrada, calcular una combinación lineal, aplicar la función sigmoide para obtener la probabilidad y, finalmente, aplicar un umbral para determinar la categoría final.

2.9.3. NOTA: Determinación de los Parámetros β

La función $z = \beta_1 x + \beta_0$ no se calcula mediante fórmulas directas como en la regresión lineal, sino mediante la optimización de la **Función de Verosimilitud** ($L(\theta)$).

Buscamos maximizar la probabilidad conjunta de observar los datos actuales:
$$L(\theta) = \prod_{i=1}^{m} P(y_i | x_i; \theta) \tag{34}$$

Para facilitar el cálculo, se aplica el logaritmo, transformándola en la **Log-Verosimilitud**, que es equivalente a minimizar la función de costo de Entropía Cruzada:
$$J(\theta) = -\sum_{i=1}^{m} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{35}$$

Los valores específicos $\beta_1 = 1{,}5$ y $\beta_0 = -4{,}5$ se obtienen tras múltiples iteraciones del **Descenso de Gradiente**, donde en cada paso se actualizan los parámetros siguiendo la regla:
$$\beta_j := \beta_j - \alpha \frac{\partial J(\theta)}{\partial \beta_j} \tag{36}$$
Donde α es la tasa de aprendizaje que controla qué tan rápido convergen los parámetros a sus valores finales.

Conclusiones

- **Propósito:** La regresión lineal resuelve problemas de *regresión* (salida continua), mientras que la logística resuelve problemas de *clasificación* (salida categórica).
- **Función de Activación:** La regresión logística introduce la función sigmoide $\sigma(z) = \frac{1}{1+e^{-z}}$ para mapear valores a probabilidades, paso inexistente en la regresión lineal.
- **Interpretabilidad:** En la lineal, β_1 es el cambio directo en y por unidad de x. En la logística, β_1 representa el cambio en el *log-odds* de la variable dependiente.
- **Robustez:** La regresión logística es generalmente más robusta ante valores atípicos en problemas de clasificación, ya que la curva sigmoide limita el impacto de valores extremos de z.

Criterio	**Regresión Lineal**	**Regresión Logística**
Variable Dependiente	Continua	Categórica (Binaria)
Función de Costo	Error Cuadrático Medio	Entropía Cruzada
Relación x/y	Lineal	Sigmoidal (Logística)
Resultado	Valor exacto	Probabilidad