Informática y comunicaciones
ies azarquiel

CURSO ESPECIALIZACIÓN INTELIGENCIA ARTIFICIAL Y BIG DATA

Capitulo 3: ALGORITMOS DE REGRESIÓN

Sebastián Rubio Valero

Septiembre 2025

Sistema aprendizaje automático

1 Algoritmo de regresión lineal simple

Estos algoritmos encuentran dentro del grupo de Aprendizaje Automático Supervisado. Existen diferentes tipos de algoritmos de regresión, por un lado están los lineales y por otro los no lineales, estos asu vez pueden ser simples o múltiples.
La regresión lineal es una técnica de análisis utilizada para prededir el valor de una variable dependiente basada en el valor de una o más variable independientes.La variable que desea predecir se denomina variable dependiente. La variable que está utilizando para predecir el valor de la otra variable se denomina variable independiente. Es importante destacar que este modelo se aplica sobre **variables continuas**.

El objetivo principal del Modelo de Regresión Lineal es comparar dos variables. Si existe una relación entre las dos variable es posible **predecir** otro valor (valores contínuos) no utilizado. Algunos ejemplos:

- Lluvias y producción de una materia
- Horas de trabajo y Productividad
- Tasas de Interés y Precio Divisa

Este algoritmo consta de los siguientes pasos:

- Definición de la función de error
- Inicialización aleatoria de los parámetros del modelo
- Definición de la tasa de aprendizaje y el número de iteraciones
- Actualización de los parámetros del modelo usando el algoritmo del gradiente descendente

Es un algoritmo de aprendizaje supervisado, por lo tanto el conjunto de datos debe de estar etiquetado. Este modelo consta de una variable dependiente, y una o más variables independientes (etiquetado). La variable dependiente es la que queremos predecir. A un valor de x le corresponde otro de y, estos punto (x,y) se distribuyen a lo largo de unos ejes de coordenadas. En la figura 1 vemos un diagrama de dispersión donde se relacionan los pesos y las alturas de una muestra de la población.

Los diagramas de dispersión son muy variados y dan lugar a diferentes modelos de regresión, en la figura 2 podemos ver algunos modelos más típicos. Analizándolos en un primer vistazo podemos decir si hay o no correlación entre las variables x e y

Peso	Altura
50	145
65	162
70	165
54	167
85	172
79	175
77	185
87	193
25	50
98	178

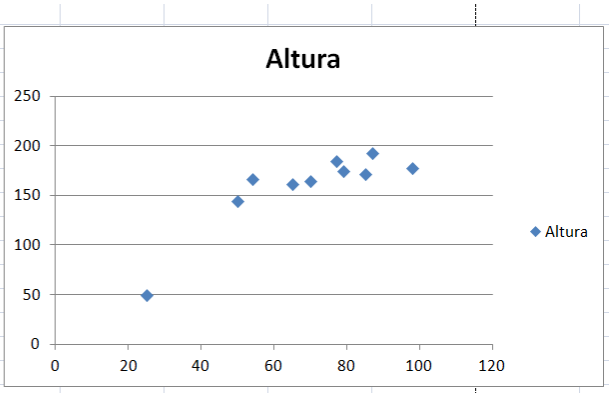

Figure 1: Diagrama de dispersión

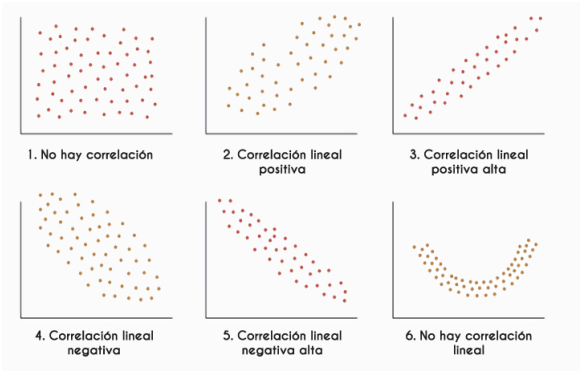

Figure 2: Diagramas de dispersión

Al final, el objetivo de crear un modelo es el de obtener una recta o curva que se ajuste lo máximo posible a los puntos del diagrama de dispersión, cuanto más ajustada esté la gráfica a los puntos mejores serán las predicciones. Empezaremos estudiando las rectas de regresión lineal simples.

La recta que queremos obtener es de la forma:

$\hat{y} = b_1 * x + b_0$ Función de hipótesis

$y = variable\ dependiente$

$x = variable\ independiente$

$b_1 = pendiente\ recta$

$b_0 = ordenada\ origen$

Partiendo de un conjuntos de datos, queremos obtener la recta que mejor ajusten eso datos, de modo que al proporcionarle un valor que no se ha contemplado, obtendremos un resultado lo más exacto posible. La recta que obtenemos va a predecir

valores, que con los datos reales va existir un error, esto error es el que vamos a tratar de minimizar los máximo posible.

Para encontrar la recta vamos utilizar el método de **mínimos cuadrados**, también se le conoce como función de coste.La elección de la función de coste va a depender del problema a resolver. Tras sucesivas iteraciones se calcula el valor de la pendiente y de la intersección con el eje y, que minimiza la suma de los errores al cuadrado (distancia a la recta). La elección de la función de coste va a depender del problema, al final del tema hay una relación de funciones de coste.

$$MSE = \sum_{i=1}^{m}(y - \hat{y})^2 = \sum_{i=1}^{m}(y - b - wx)^2 \tag{1}$$

$$\hat{y} = wx + b \tag{2}$$

Donde:

- $\hat{y}$: valor predicho
- x: variable independiente
- w: pendiente de la recta
- b: intercepto (bias)

Al cálculo de los parámetros w y b se le conoce como entrenamiento. Para empezar a estimar los mejores valores de w b partimos de una función hipótesis, de una buena elección de esta función va a depender la velocidad de convergencia del modelo.

El objetivo es minimizar la suma de los errores cuadráticos de la función hipótesis. Como se puede observar es una función parabólica por lo que podemos encontrar el mínimo de esta función tras calcular su primera derivada e igualándola a cero

Cualquier predicción que hagamos con la recta de regresión tendrá un error. Lo ahora vamos hacer acotar en un intervalo dicho error en pronóstico, se le conoce como error de pronóstico.

Una forma sencilla de entrenar una I.A para la regresión lineal, es la de proponer en un primer paso un recta de regresión (función de hipótesis) y luego de forma iterativa optimizarla.

$$J(w, b) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(wx_i + b - y_i)^2 \tag{3}$$

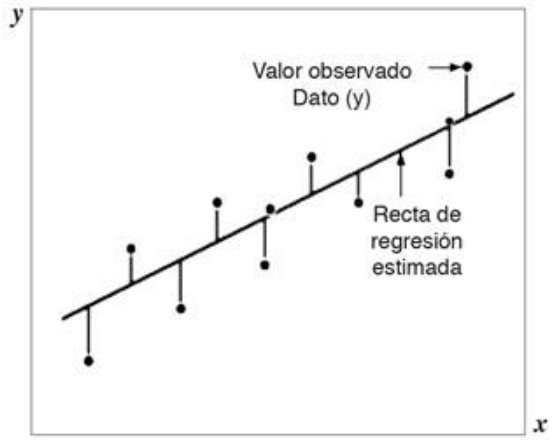

Figure 3: Residuo

La función de optimización, que más se utilizada es el gradiente descendente. Lo que realmente se busca es el punto en el que la primera derivada es cero. Como podemos observar nuestra función de error es una parábola, las cuales siempre podemos encontrar un mínimo. Cuando el problema consta de más de una variable usaremos las derivadas parciales

$$\frac{\partial J}{\partial w} = \frac{2}{n}\sum_{i=1}^{n} x_i(wx_i + b - y_i) \tag{4}$$

$$\frac{\partial J}{\partial b} = \frac{2}{n}\sum_{i=1}^{n} (wx_i + b - y_i) \tag{5}$$

Una vez que hemos calculado las derivadas parciales con las ecuaciones anteriores, podemos actualizar los parámetros w y b. Con estos nuevos parámetros vamos a obtener una nueva recta regresión en la que el error cuadrático medio será menor que en la anterior recta.

$$w^{i+1)} = w^i - \alpha\frac{\partial J}{\partial w} \tag{6}$$

$$b^{i+1)} = b^i - \alpha\frac{\partial J}{\partial b} \tag{7}$$

Donde α es la tasa de aprendizaje, es un número pequeño y positivo que determina cuanto cambia los parámetros en cada iteración. La elección de este factor va afectar a la convergencia del modelo se la sguiente forma:

Para que se entienda podemos poner la analogía de bajar por una montaña.
No hay una fórmula que nos diga que valor de α se puede elegir, pero existen métodos

Valor de α	**Comportamiento del modelo**
Muy pequeño (α=0.0001)	Aprende muy lento (muchas iteraciones)
Muy grande (α=1.0)	Puede divergir (no converge, saltos muy grandes)
Apropiado (α=0.01)	Converge rápido y estable al mínimo del error

Table 1: Notas matemáticas y lengua

Matemáticas	**Lengua**
7	4
5	5
3	6
6	6
8	5
3	2
8	6

Table 2: Notas matemáticas y lengua

que nos pueden ayudar a su elección, como por ejemplo la validación cruzada, la tasa de aprendizaje adaptativa, reducción de la tasa de aprendizaje por épocas. Al implementar el algoritmo siguiendo las ecuaciones mostradas anteriormente podemos de forma iterativa minimizar la función de error para encontrar la recta que mejor se ajuste a los datos.
El procedimiento descrito anteriormente, a través del cuál se calcular de forma automática los parámetros ω y b usando el método de gradiente descendente, se conoce como proceso de entrenamiento, es decir realizar sucesivos cálculos de los parámetros hasta que el modelo converja. La mejor forma de entenderlo en estudiando un sencillo ejemplo.

EJEMPLO: En la siguiente tabla la variable independiente x hace referencia a las notas de matemáticas, y la variable dependiente y hace referencia a las notas de lengua. Queremos encontrar un modelo que nos diga la nota de lengua sabiendo la nota de matemáticas.

Los cálculos de la primera iteración.

PASO 1: Inicializar
$w = 0, b = 0, \alpha = 0.01$, *Número de iteraciones = 1.*

PASO 2: Calcular la predicción

$$\hat{y}_i = wx_i + b = 0.x_i + 0 = 0 \tag{8}$$

x	y	$\hat{y}$	Error
7	4	0	-4
5	5	0	-5
3	6	0	-6
6	6	0	-6
8	5	0	-5
3	2	0	-2
8	6	0	-6

Table 3: Cálculo del error en la primera iteración

$$Error = \hat{y}_i - y_i \tag{9}$$

PASO 3: Calcular de los gradientes

$$\frac{\partial J}{\partial w} = 2/7[7((0*7+0)-4)+5((0*5+0)-5)+3((0*3+0)-6)+6(0*6+0)-6)+ \tag{10}$$

$$+8((0*8+0)-5)+3((0*3+0)-2)+8((0*8)+0)-6] \tag{11}$$

$$\frac{\partial J}{\partial w} = 2/7[7(-4)+5(-5)+3(-6)+6(-6)+8(-5)+3(-2)+8(-6)] \tag{12}$$

$$\frac{\partial J}{\partial w} = 2/7[-28-25-16-36-40-6-48] = -56,85 \tag{13}$$

$$\frac{\partial J}{\partial b} = 2/7[-4-5-6-6-5-2-6] = -9,71 \tag{14}$$

PASO 4: Actualizar parámetros

$$w = 0 - 0.01(-56,85) = 0,5685$$
$$b = 0 - 0,01(-6,67) = 0,0667$$

PASO 4: Modelo en la primera iteración

$$\hat{y}_i = 0,5685x_i + 0,0667 \tag{15}$$

EJERCICIO 1.1
Partiendo del ejemplo anterior realizar otras dos iteraciones (w=1,w=2). Comprueba tus resultados con otros compañeros

EJERCICIO 1.2
Realizar el ejercicio anterior hasta 4 iteraciones con una tasa de aprendizaje α=0.1. Pero ahora reserva las dos últimas filas para estimar el modelo. Comprueba los resultados con otros compañeros

EJERCICIO 1.3
Observa las diferencias entre los modelos del ejercicio 1.1 y del ejercicio 1.2

Lo que hemos hecho en este ejercicio es tomar todos nuestras datos y utilizarlos para el entrenamiento del modelo. En un caso real tenemos que dividir nuestros ejemplos en dos grupos, el de entrenamiento, y el de comprobación. Esto lo estudiaremos con un ejercicio práctico en python.

Estos ejemplos también podemos verlos con una hoja de calculo excel

MÉTRICAS PARA LOS PROBLEMAS DE REGRESIÓN

- Error cuadrático medio. Promedio de errores, sensible a outliers
- Raíz del error cuadrático medio. Promedio de errores no sensible a outliers
- Error absoluto medio. Promedio de valores absolutos, no sensibles outliers, pero es lento
- Coeficiente de determinación. Toma valores entre 0 y 1 es el mejor. Varianza de la variable dependiente
- Error Absoluto Medio Porcentual, Eror promedio en términos porcentuales
- Error Cuadrático Medio Logarítmico. Promedio sobre algoritmos

Una vez que hemos visto la parte teórica, a continuación vamos a ver dos ejemplos. Los dos cuadernos de notebook, estan en el aula virtual

- Regresion-lineal-simple.ipynb

- Otro ejemplo

Por último se propone un ejercicio. El dataset correspondiente se encuentra en el aula virtual

2 Regresión lineal mútiple

En estos modelos tenemos más de una variable independiente y una variable dependiente. En estos tipos de regresión obtenemos un plano si son dos las variables independientes, si son más de dos entonces estamos hablando de un hiperplano.

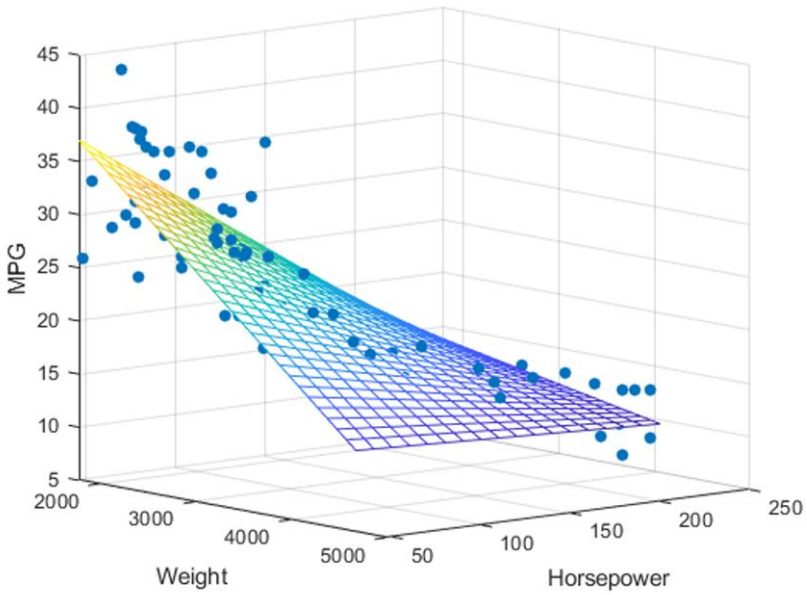

Figure 4: Enter Caption

La ecuación del plano viene dada como:

$$\hat{y} = w_1 x_1 + w_2 x_2 + b \tag{16}$$

Y el error que queremos minimizar es:

$$MSE = \sum_{i=1}^{m} (y - \hat{y})^2 = \sum_{i=1}^{m} (y - (b - w_1 x_1 - w_2 x_2))^2 \tag{17}$$

- $\hat{y}$: valor predicho
- x_1, x_2: variable independiente
- w_1, w_2: pendiente de la recta
- b: sesgo (bias)

Para estudiar este modelo de regresión vamos a ver un ejemplo
EJEMPLO:Regresión Múltiple Datos
x_1 =horas dormidas por dia x_2 =horas de estudio por día y =nota PAU

x_1	x_2	y
7	5	13
8	4	13,5
8	7	14

Table 4: Tabla de datos

PASO 1: Inicialización
$w_1 = 0, w_2 = 0, b = 0, \alpha = 0.01$, *Número de iteraciones = 1.*

PASO 2: Calcular las primeras predicciones

$$y^i = w_1 x_1^i + w_2 x_2^i + b = 0 \tag{18}$$

x_1	x_2	y	$\hat{y}$	Error $\hat{y} - y$
7	5	13	0	-13
8	4	13,5	0	-13,5
8	7	14	0	-14

Table 5: Sample Data Table with Errors

PASO 3:Gradientes

$$\frac{\partial J}{\partial x_1} = \frac{2}{n}\sum_{i=1}^{n} x_{1i}(\hat{y}_i - y_i) = \frac{2}{3}\left(7 \cdot (-13) + 8 \cdot (-13,5) + 8 \cdot (-14)\right) = \frac{2}{3}(-311) = -207,33 \tag{19}$$

$$\frac{\partial J}{\partial x_2} = \frac{2}{n}\sum_{i=1}^{n} x_{2i}(\hat{y}_i - y_i) = \frac{2}{3}\left(5 \cdot (-13) + 4 \cdot (-13,5) + 7 \cdot (-14)\right) = \frac{2}{3}(-217) = -144,6 \tag{20}$$

$$\frac{\partial J}{\partial b} = \frac{1}{2}\sum_{i=1}^{n}(\hat{y}_i - y_i) = \frac{1}{2}(-13 - 13,5 - 14) = \frac{1}{2}(-40,5) = -20,25 \tag{21}$$

PASO 4: Actualización de parámetros
$w_1 := 0 - 0.01(-207,33) = 2,0733$
$w_2 := 0 - 0.01(-144,6) = 1,446$
$b := 0 - 0.01.(-20,25) = 0,2025$

Ejercicio

Se deja como ejercicio la realización de otros 3 o 4 iteraciones (usar un hoja de calculo excel). Realizar de nuevo el ejercicio cambiando el parámetro de la tasa de aprendizaje α

2.1 Importancia de la tasa de aprendizaje

2.2 Ecuación normal

Para obtener el modelo existe una solución cerrada utilizando la matriz de características por vectores

$$\mathbf{X} = \begin{pmatrix} 1 & x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(n)} \\ 1 & x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(n)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N^{(1)} & x_N^{(2)} & \cdots & x_N^{(n)} \end{pmatrix} \tag{22}$$

Necesitamos también la matriz de los valores de la variable objetivo

$$\mathbf{X} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \tag{23}$$

Con la *Ecuación normal* podemos obtener los parámetros

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{24}$$

Para aplicar esta fórmula tenemos que estar seguros que existe la inversa, es decir el $det(\mathbf{X}^T\mathbf{X})^{-1} \neq 0$.
Tiene la ventaja entre otra que no es necesario el factor de convergencia.

2.3 Otras funciones de coste

2.3.1 Error Cuadrático Medio (MSE)

Esta es, con diferencia, la función de costo más utilizada y la que se enseña habitualmente en la derivación de la regresión lineal por Mínimos Cuadrados Ordinarios (OLS).

Fórmula: $J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ Donde:

$J(\theta)$ es la función de costo.

n es el número de observaciones.

y_i es el valor real.

$\hat{y}_i$ es el valor predicho $(\theta_0 + \theta_1 x_1 + ... + \theta_n x_n)$.

¿Por qué es tan popular?

Convexidad: La función MSE es convexa respecto a los parámetros θ. Esto garantiza que el algoritmo de optimización (como el Descenso de Gradiente) encuentre el mínimo global y no se quede atascado en un mínimo local.

Diferenciabilidad: Es fácilmente diferenciable en todos sus puntos, lo que permite calcular el gradiente de manera eficiente, necesario para el Descenso de Gradiente.

El gradiente de MSE es: $\nabla J(\theta) = -\frac{2}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \cdot x_i$

Desventaja: Es muy sensible a los valores atípicos (outliers). Como los errores se elevan al cuadrado, un solo outlier muy lejano inflará enormemente el costo, dominando el proceso de optimización y sesgando los resultados.

2.3.2 Error Absoluto Medio (MAE)

Esta función de costo es una alternativa robusta al MSE.

Fórmula: $J(\theta) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

Ventaja principal:

Robustez a Outliers: Al usar el valor absoluto en lugar del cuadrado, los errores grandes no se penalizan de manera desproporcionada. El modelo resultante será menos sensible a puntos de datos anómalos.

Desventajas:

No diferenciabilidad en cero: La derivada de la función valor absoluto no está definida en el punto cero. Esto puede causar problemas menores en la optimización, que se suelen resolver usando subgradientes.

Múltiples soluciones: La función de costo MAE puede tener múltiples conjuntos de parámetros que conduzcan al mismo valor mínimo de costo, es decir, puede no tener una solución única.

2.3.3 Error Cuadrático Medio de Huber (Huber Loss)

Esta es una función de costo híbrida diseñada específicamente para combinar lo mejor del MSE y el MAE. Es menos sensible a outliers que el MSE pero mantiene ciertas propiedades de diferenciabilidad.

Concepto: Utiliza una transición suave entre un comportamiento cuadrático (como el MSE) para errores pequeños y un comportamiento lineal (como el MAE) para errores grandes. Un parámetro delta (δ) define el umbral donde ocurre esta transición.

Fórmula por partes: $L_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{para } |y - \hat{y}| \leq \delta \\ |y - \hat{y}| - \frac{1}{2}\delta^2 & \text{para } |y - \hat{y}| > \delta \end{cases}$

Ventajas:

Lo mejor de ambos mundos: Es diferenciable en todas partes (incluyendo en $|y - \hat{y}| = \delta$) y es robusta a outliers.

Eficiencia estadística: Para errores pequeños, se comporta como MSE, lo que often resulta en estimadores más eficientes.

Desventaja: Introducir el hiperparámetro δ añade complejidad, ya que hay que elegir su valor adecuado (usualmente mediante validación cruzada).

2.4 Métricas para evaluar los algoritmos

Las métricas para evaluar algoritmos de regresión lineal se dividen en dos categorías: aquellas que miden el **error** (qué tan equivocadas son las predicciones) y aquellas que miden la **bondad** del ajuste (qué tan bueno es el modelo explica la variabilidad de los datos). Medir el rendimiento de un modelo se realiza sobre le conjunto de datos de validación, no sobre el conjunto de datos con el que hemos entrenado el modelo.

2.4.1 Error Cuadrático Medio (MSE):De error

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{25}$$

- **Ventajas:**
  - Penaliza errores grandes (útil para identificar *outliers*).
  - Diferenciable (ideal para optimización con gradiente descendente).
- **Desventajas:**
  - Sensible a *outliers* (puede inflarse el error).
  - No está en la misma unidad que la variable objetivo.

2.4.2 Raíz del Error Cuadrático Medio (RMSE):Error

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{26}$$

- **Ventajas:**
  - Misma unidad que la variable objetivo (interpretación intuitiva).
- **Desventajas:**
  - Igualmente sensible a *outliers*.

2.4.3 Error Absoluto Medio (MAE):Error

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{27}$$

- **Ventajas:**
  - Robusto frente a *outliers*.
  - Interpretación directa (error promedio en unidades originales).
- **Desventajas:**
  - No es diferenciable en cero (problemas en optimización).

2.4.4 Coeficiente de Determinación (R^2):Bondad

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{28}$$

- **Ventajas:**
  - Interpretación intuitiva (% de varianza explicada por el modelo).
  - Normalizado entre 0 y 1 (fácil comparación entre modelos).
- **Desventajas:**
  - Puede ser engañoso con datos no lineales.
  - Aumenta artificialmente al añadir más variables.

2.4.5 R^2 Ajustado:Bondad

$$R^2_{\text{ajustado}} = 1 - \left(\frac{(1 - R^2)(n - 1)}{n - p - 1} \right) \tag{29}$$

- **Ventajas:**
  - Penaliza la inclusión de variables irrelevantes (evita sobreajuste).
- **Desventajas:**
  - Menos interpretable que R^2 estándar.

2.4.6 Error Porcentual Absoluto Medio (MAPE)

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{30}$$

- **Ventajas:**
  - Interpretación en porcentaje (fácil comunicación).
- **Desventajas:**
  - Indefinido si $y_i = 0$.
  - Penaliza más errores en valores pequeños.

2.4.7 Resumen

Métrica	Usar Cuando...	Evitar Cuando...
MSE	Penalizar errores grandes.	Hay muchos *outliers*.
RMSE	Error en unidades originales.	*Outliers* problemáticos.
MAE	Robustez frente a *outliers*.	Necesitas diferenciabilidad.
R^2	Medir varianza explicada.	Hay sobreajuste.
R^2 Ajustado	Comparar modelos con más variables.	Modelos simples.
MAPE	Errores en porcentaje.	Valores cercanos a cero.

Table 6: Guía rápida para selección de métricas.

Para evaluar completamente un modelo de regresión lineal, es recomendable usar una métrica de error (RMSE o MAE) y una métrica de bondad de ajuste (R^2 Ajustado).

RMSE te dirá "cuánto" se equivoca en promedio.

R^2 Ajustado te dirá "qué tan bien" se ajusta a los datos, considerando su complejidad.

Ninguna métrica por sí sola cuenta la historia completa. Siempre debes analizarlas en conjunto y, crucialmente, visualizar los residuos para detectar patrones que las métricas puedan estar ocultando.

3 REGRESIÓN POLINOMIAL

El algoritmo de regresión polinomial es una extensión del algoritmo lineal simple que modela la relación entre una variable independiente (X) y una variable dependiente (y) como un polinomio de grado n.

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + ... + \beta_n * x_n \tag{31}$$

Para el caso de una variable independiente :
$y = b + a_1x_1$ su correspondiente polinomial sería $y = b + a_1x_1 + a_2x_1^2$
Con dos variables independientes:
$y = b + a_1x_1 + a_2x_2$ su correspondiente polinomial
$y = b + a_1x_1 + a_2x_1^2 + a_3x_2 + a_4x_2^2 + a_5x_1x_2$, es decir
$y = b + a_1X_1 + a_2X_2 + a_3X_3 + a_4X_4 + a_5X_5$

Con esta nueva ecuación ya podemos aplicar el modelo de regresión lineal que hemos estado estudiando.
Este modelo al tener más parámetros, tiene tendencia al sobreajuste
El alumno se preguntará y ¿si tengo uno multidimensional y polinomial?, en estos casos es más complicado, se podría utilizar el algoritmo de regresión, pero es mejor solucionarlo por otros técnicas de machine learning.

4 HIPERPARÁMETROS DEL ESQUEMA DE APRENDIZAJE

Los hiperparámetros del esquema de aprendizaje (o learning schedule hyperparameters) son valores de configuración que controlan cómo evoluciona la tasa de aprendizaje (η o LR) de un modelo de machine learning (especialmente redes neuronales) a lo largo del entrenamiento. En lugar de usar una tasa de aprendizaje fija (estática), un esquema de aprendizaje la ajusta dinámicamente para mejorar la estabilidad, acelerar la convergencia y lograr un mejor rendimiento del modelo.

4.1 Conceptos Fundamentales

Los siguientes conceptos son esenciales en algunos algoritmos de aprendizaje automático

- Tasa de Aprendizaje Incial(η_0 o $LR_{initial}$):Es el valor del LR al comienzo del entrenamiento, antes de que el esquema de aprendizaje comience a modificarlo.
- Épocas (n_{epochs}): El número total de veces que el algoritmo recorrerá todo el conjunto de datos de entrenamiento. El esquema de aprendizaje a menudo depende de este número
- Tasa de Dacaimiento (Dacay Rate): En los esquemas de decaimiento escalonado, es el número de épocas después de las cuales se reduce el LR por un factor fijo
- Intervalo de Dacaimiento (Step Size o Epochs Drop)

4.2 Tipos Comunes de Esquemas de Aprendizaje y sus Hiperparámetros

Cada esquema de aprendizaje define una función para actualizar el LR y, por lo tanto, tiene sus propios hiperparámetros específicos que deben ajustarse (sintonizarse):

4.2.1 Decaimiento Basado en el Tiempo (Time-Based Decay)

El LR disminuye continuamente con el tiempo (por iteración o época).
Fórmula: $\eta_t = \frac{\eta_0}{1+k \cdot t}$
Hiperparámetros Clave:

- η_0 Tasa de aprendizaje inicial
- k (Decaimiento), factor que controla qué tan rápido disminuye el LR.

4.2.2 Decaimiento Escalonado (Step Decay)

El LR se reduce por un factor fijo después de un número predefinido de épocas.
Fórmula: $\eta_t = \eta_0 \cdot (\text{factor de decaimiento})^{\lfloor t/\text{intervalo de decaimiento} \rfloor}$
Hiperparámetros Clave:

- η_0: Tasa de aprendizaje inicial.
- Factor de Decaimiento (Drop Factor): El valor por el que se multiplica el LR (ej: 0.5 para reducirlo a la mitad).

- Intervalo de Decaimiento (Step Size):La frecuencia (en épocas) con la que se aplica la reducción.

4.2.3 Decaimiento Exponencial (Exponential Decay)

El LR disminuye exponencialmente con el tiempo.
Fórmula:$\eta_t = \eta_0 \cdot e^{-k \cdot t}$ Hiperparámetros Clave:

- η_0:Tasa de aprendizaje inicial.
- k (Tasa de Decaimiento):Factor que determina la rapidez de la disminución exponencial.

4.2.4 Recocido Cosenoidal (Cosine Annealing)

El LR sigue una curva cosenoidal, comenzando alto, disminuyendo hasta un mínimo y a veces volviendo a "reiniciarse" periódicamente. Es muy popular en el deep learning moderno.
Hiperparámetros Clave

- η_{min}, η_{max}:Los valores mínimo y máximo a los que oscilará el LR.
- T_{max}:El número de iteraciones o épocas en medio ciclo cosenoidal (o el periodo total de un ciclo completo).

4.2.5 Tasa de Aprendizaje Cíclica (Cyclical Learning Rate - CLR)

El LR oscila entre un valor mínimo y uno máximo siguiendo un patrón (a menudo triangular o cosenoidal) en lugar de simplemente disminuir. Esto ayuda al modelo a escapar de los mínimos locales.
Hiperparámetros Clave:

- LR_{min}, LR_{max}:Los límites inferior y superior del LR.
- Step Size:El número de iteraciones en medio ciclo.

4.3 La Importancia de los Esquemas de Aprendizaje

El problema de la tasa de aprendizaje fija es un dilema, hoy en día hay muchos estudios sobre la tasa de aprendizaje, sobre todo por el coste que supone entrenar algunos modelos: Si LR es alto al inicio, ayuda a converger rápido, pero oscila o no llega al mínimo exacto al final. Por el contrario, si LR Bajo al inicio, converge de forma

estable, pero tarda mucho y puede quedar atrapado en un mínimo local subóptimo.

El esquema de aprendizaje resuelve esto:
Inicio (Fase de Exploración): Se usa un LR alto para moverse rápidamente hacia la región general del mínimo (Fase de Warmup en algunos esquemas).
Final (Fase de Ajuste Fino): Se usa un LR bajo para estabilizar la convergencia, evitar oscilaciones y "afinar" los pesos para alcanzar el mínimo más profundo.

5 FUENTES

Aprende Machine Learning con sckiti-learn O'REILLY (ANAYA). Apuntes de la UNED.