Regresión y Selección de Modelos en Machine Learning

Contenido

Regresión vs Clasificación

Clasificación

- Predecir **categorías discretas**
  - ¿Es spam o no spam?
  - ¿Qué tipo de componente es?

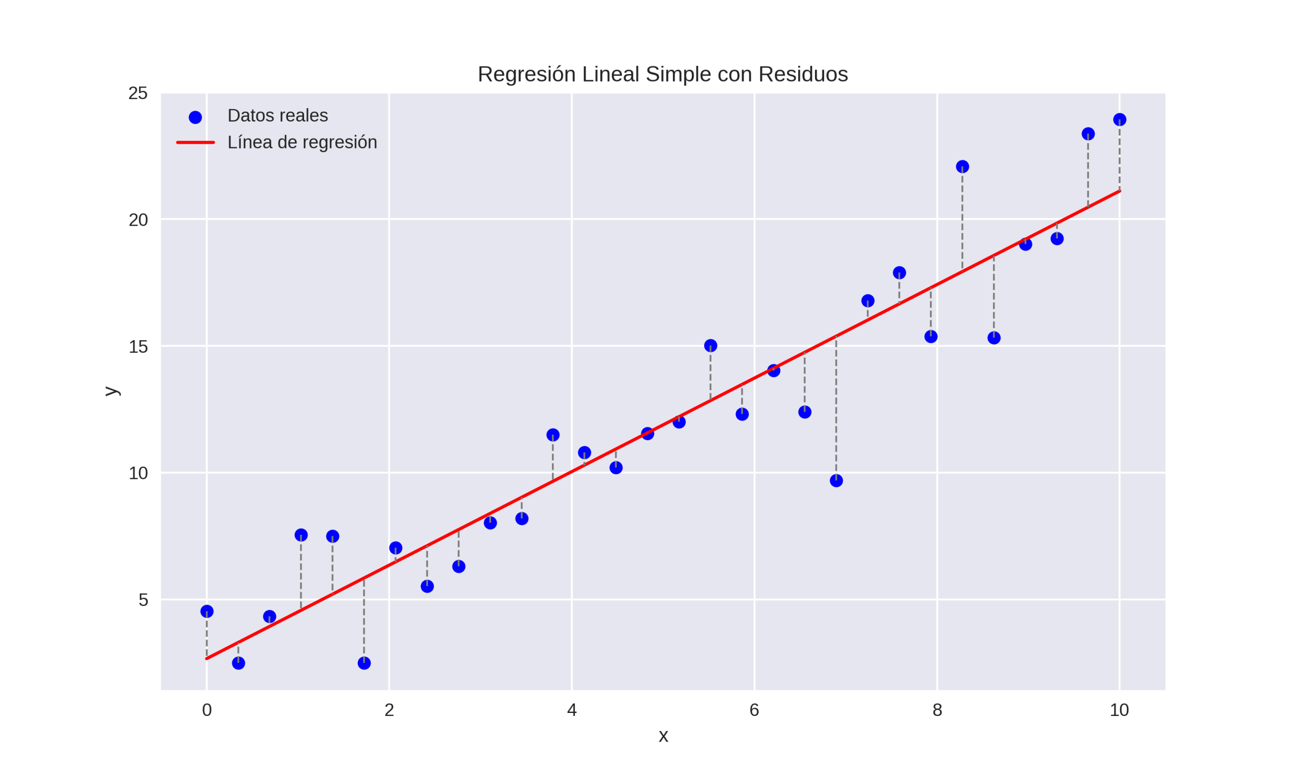
Regresión

- Predecir **valores continuos**
  - ¿Cuál será la temperatura?
  - ¿Cuánto voltaje necesitamos?
  - ¿Cuál es la resistencia del material?

Regresión Lineal Simple

Concepto Fundamental

- Encontrar la recta que mejor se ajusta a los datos
- Minimizar distancia entre los puntos reales y la recta
- Relación lineal entre una variable independiente y una dependiente



Modelo Matemático

$$y = \beta_0 + \beta_1 x + \epsilon$$

Donde:

- y: variable dependiente (respuesta)
- x: variable independiente (predictor)
- β_0: intercepto (valor de y cuando $x = 0$)
- β_1: pendiente (cambio promedio en y por unidad de x)
- ϵ: error aleatorio (ruido)

Supuestos del Modelo

1. **Linealidad**: Relación lineal entre x e y
2. **Independencia**: Observaciones independientes
3. **Homocedasticidad**: Varianza constante del error
4. **Normalidad**: Errores normalmente distribuidos

Método de Mínimos Cuadrados

Encontrar los valores de β_0 y β_1 que minimizan la suma errores cuadráticos

Suma de los Errores Cuadráticos (SSE - *Sum of Squared Errors*)

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

Donde:

- n: número de observaciones
- y_i: valor real observado
- $\hat{y}_i = \beta_0 + \beta_1 x_i$: valor predicho

Solución Analítica

Los coeficientes óptimos son:

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \quad \text{e} \quad\quad \beta_0 = \bar{y} - \beta_1\bar{x}$$

Donde:

- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (media de x)
- $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ (media de y)

Métricas de Evaluación del Modelo de Regresión

- Una predicción numérica, en general, es poco probable que sea exactamente correcta, pero puede estar cerca o lejos del valor verdadero.
- Las mediciones numéricas provienen de una distribución con cierto grado de incertidumbre, el "error".

Métricas para Regresión

1 Error Cuadrático Médio (MSE - *Mean Squared Error*)

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

- Promedio de errores cuadráticos
- Penaliza más los errores grandes
- Expresado con el cuadrado de las unidades de y

2 Raiz del Error Cuadrático Médio (RMSE - *Root Mean Squared Error*)

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

- Más intuitiva que MSE
- Está en la misma unidad de medida que que el valor real y
- Penaliza un poco menos los errores grandes en relación al MSE

3 Error Médio Absoluto (MAE - *Mean Absolute Error*)

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

- Promedio de errores absolutos
- Mismas unidades que y
- Menos sensible a errores grandes (***outliers***)
- Más estable y confiable cuando hay datos ruidosos o errores extremos.

4 R^2 Coeficiente de Determinación (R-squared - *Coefficient of Determination*)

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

Mide la fracción de la variabilidad total que el modelo logra explicar

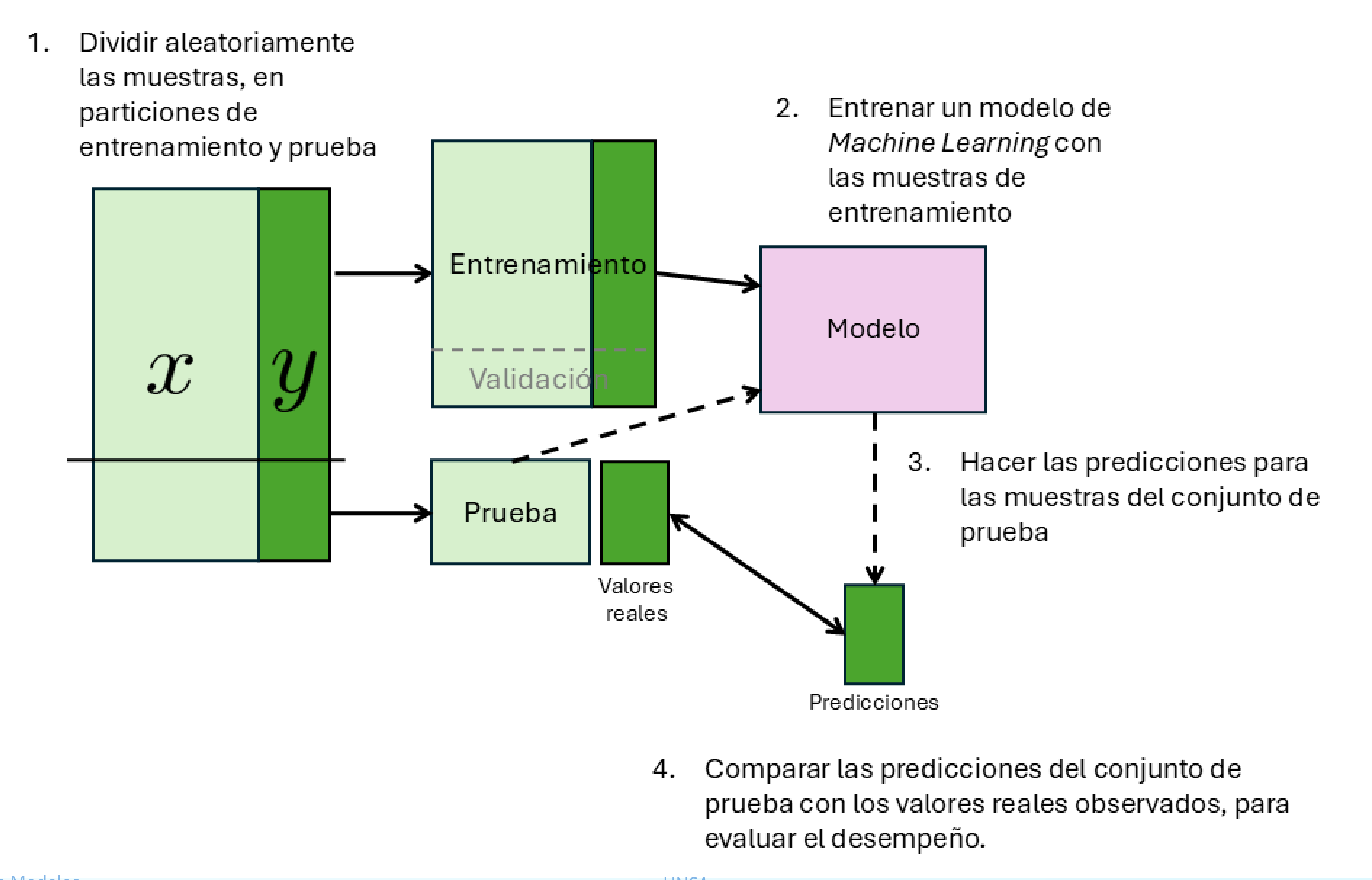
Valor R^2	**Qué significa**
1	Explica toda la variabilidad de los datos (perfecto ajuste)
0	No explica mejor que la media (predicción constante).
entre 0 y 1	Explica parte de la variabilidad
< 0	El modelo es peor que predecir siempre la media

Métodos de Selección de Modelos

- El error en el conjunto de entrenamiento no es indicativo del error del modelo cuando se aplica a nuevos datos (prueba).
- Para la estimación del error en los datos de prueba, se usa la validación cruzada (CV- ***Cross Validation***).
- Validación cruzada es cualquier técnica que evalúe el modelo usando datos no vistos, separados de los datos de entrenamiento.
- Los dos métodos más comúnmente utilizados para la validación cruzada son el método de **partición simple** y la **validación cruzada k-fold**.

Método de Partición Simple (Holdout Method)

- Utiliza dos particiones de los datos, para entrenamiento y prueba. Solo se utiliza la partición de entrenamiento para ajustar el modelo, y únicamente la partición de prueba para evaluar su precisión.
- En la práctica, se suele separar para prueba entre 20% a 40% de los datos.


1. Dividir aleatoriamente las muestras, en particiones de entrenamiento y prueba
x
y
Entrenamiento
Validación
Prueba
Valores reales
2. Entrenar un modelo de *Machine Learning* con las muestras de entrenamiento
Modelo
3. Hacer las predicciones para las muestras del conjunto de prueba
Predicciones
4. Comparar las predicciones del conjunto de prueba con los valores reales observados, para evaluar el desempeño.

Código con sklearn: Holdout 70/30

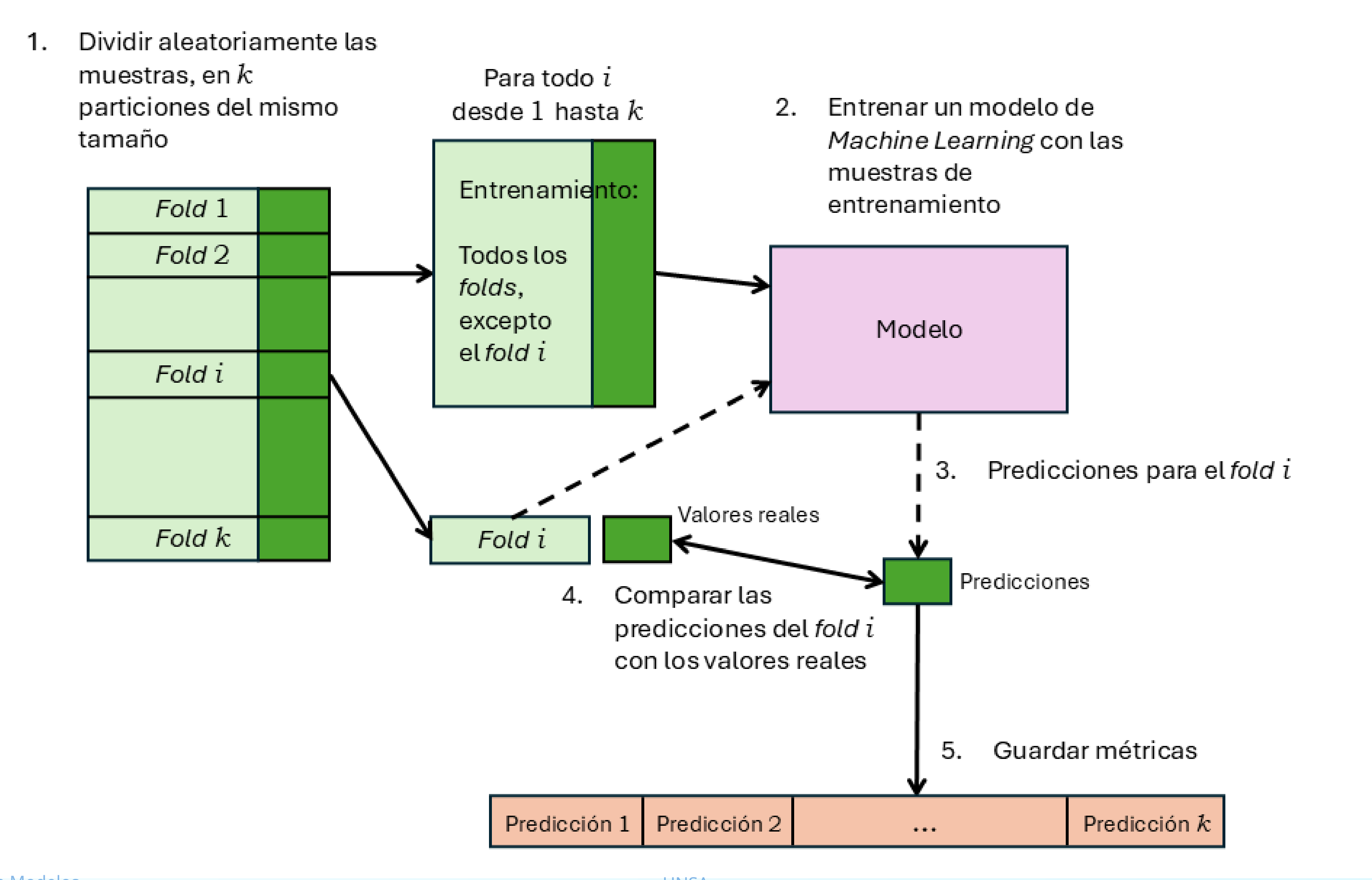
```python
# División Holdout 70% entrenamiento, 30% prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Entrenar el modelo de regresión lineal
modelo = LinearRegression()
modelo.fit(X_train, y_train)

# Evaluar el modelo
y_pred = modelo.predict(X_test)
print("Coeficiente (pendiente):", modelo.coef_[0][0])
print("Intersección:", modelo.intercept_[0])
print("Error cuadrático medio (MSE):", mean_squared_error(y_test, y_pred))
print("R² Score:", r2_score(y_test, y_pred))
```

Método de Validación Cruzada k-fold (k-fold Cross-Validation)

- Divide aleatoriamente los datos en k particiones, llamados **folds**
- Para cada **fold**, se entrena un modelo usando todos los datos excepto los del **fold** actual, y luego se utiliza ese modelo para generar predicciones sobre los datos del **fold** que se dejó fuera.
- Después de haber procesado los k **folds**, se tienen predicciones para todos los datos, y se comparan con los valores reales para calcular métricas como MSE, RMSE o R^2. El promedio de esas k métricas da una estimación más robusta y generalizable del desempeño del modelo en datos no vistos.


1. Dividir aleatoriamente las muestras, en k particiones del mismo tamaño
Fold 1
Fold 2
Fold i
Fold k
Para todo i desde 1 hasta k
Entrenamiento:
Todos los folds, excepto el fold i
2. Entrenar un modelo de Machine Learning con las muestras de entrenamiento
Modelo
3. Predicciones para el fold i
Fold i
Valores reales
Predicciones
4. Comparar las predicciones del fold i con los valores reales
5. Guardar métricas
Predicción 1
Predicción 2
...
Predicción k

Regresión Lineal Simple

Ejemplo: Sensor de Temperatura

- Proyecto de un modelo matemático que relaciona el voltaje del sensor (V) con la temperatura real (°C) de algun equipamento.
- Útil para convertir voltaje a temperatura

Datos de Calibración

Lectura Sensor (V)	Temperatura Real (°C)
0.1	5
1.2	25
2.4	50
3.6	75
4.8	100

Cálculo Paso a Paso

1. Calcular Medias

- $\bar{x} = \frac{0.1+1.2+2.4+3.6+4.8}{5} = \frac{12.1}{5} = 2.42V$
- $\bar{y} = \frac{5+25+50+75+100}{5} = \frac{255}{5} = 51^\circ C$

2. Calcular β_1 (Pendiente)

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{283.4}{13.928} = 20.35 \text{ V}/^\circ\text{C}$$

3. Calcular β_0 (Intercepto)

$$\beta_0 = \bar{y} - \beta_1 \bar{x} = 51.0 - (20.35)(2.42) = 1.76 \text{ V}$$

Ecuación Final

$$y = 1.76 + 20.35x$$

Interpretación

- **Intercepto (1.76V)**: Lectura del sensor a 0°C
- **Pendiente (20.35 V/°C)**: El sensor aumenta 20.35V por cada grado Celsius
- **Aplicación**: Para convertir voltaje a temperatura: $T = \frac{V - 1.76}{20.35}$

Ejemplo interactivo

Regresión Lineal Múltiple

Extensión de la regresión simple para múltiples variables independientes o predictoras. Permite modelar relaciones complejas donde la variable dependiente está influenciada por varios factores.

Modelo Matemático

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \epsilon$$

Donde:

- y: variable dependiente
- $x_1, x_2, \dots, x_p$: variables independientes (predictoras)
- β_0: intercepto, $\beta_1, \beta_2, \dots, \beta_p$: coeficientes de regresión, ϵ: error aleatorio

Forma Matricial

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

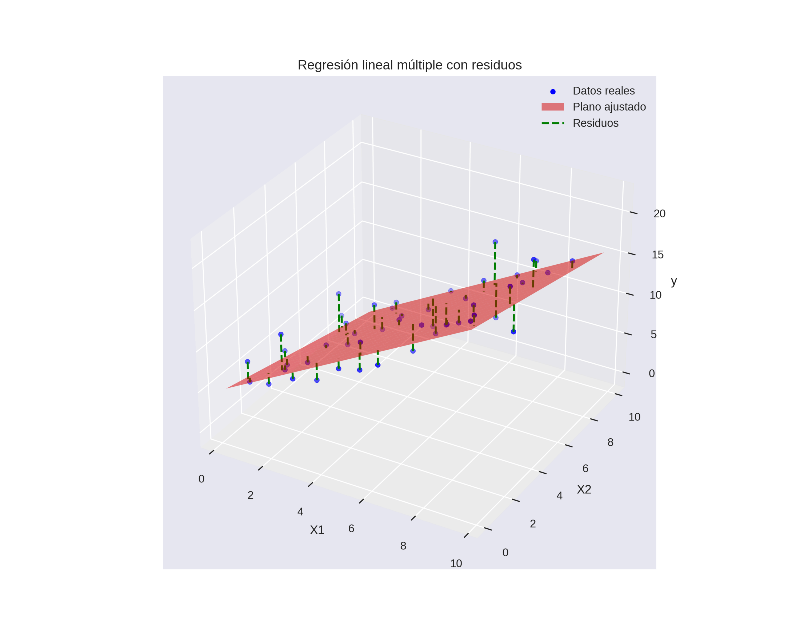
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Donde: $\mathbf{y}_{n\times 1}$: vector de respuestas

$\mathbf{X}_{n\times(p+1)}$: matriz de diseño

$\boldsymbol{\beta}_{(p+1)\times 1}$: vector de coeficiente

$\boldsymbol{\epsilon}_{n\times 1}$: vector de errores



Solución por Mínimos Cuadrados

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

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

- β_0 : (intercepto): Valor predicho de y cuando todas las entradas son cero ($x_1 = x_2 = 0$)
- β_1 : Es el efecto marginal de x_1 sobre y, manteniendo x_2 constante: Cuánto cambia y en promedio si se aumenta x_1 en una unidad, dejando x_2 fijo.
- $\beta 2$: Lo mismo, para x_2: cuánto cambia y por unidad de x_2, manteniendo x_1 constante.

Ventajas sobre Regresión Simple

1. **Mayor precisión**: Incluye más información
2. **Control de confusión**: Aísla el efecto de cada variable
3. **Interacciones**: Puede modelar efectos combinados
4. **Flexibilidad**: Adaptable a problemas complejos

Regresión Polinomial

Extensión de la regresión lineal que permite modelar relaciones no lineales mediante potencias de la variable independiente.

Modelo Matemático

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_d x^d + \epsilon$$

Donde:

- d: grado del polinomio
- $x, x^2, \ldots, x^d$: términos polinomiales
- $\beta_0, \beta_1, \ldots, \beta_d$: coeficientes

Grados Comunes

- **Grado 1**: Lineal

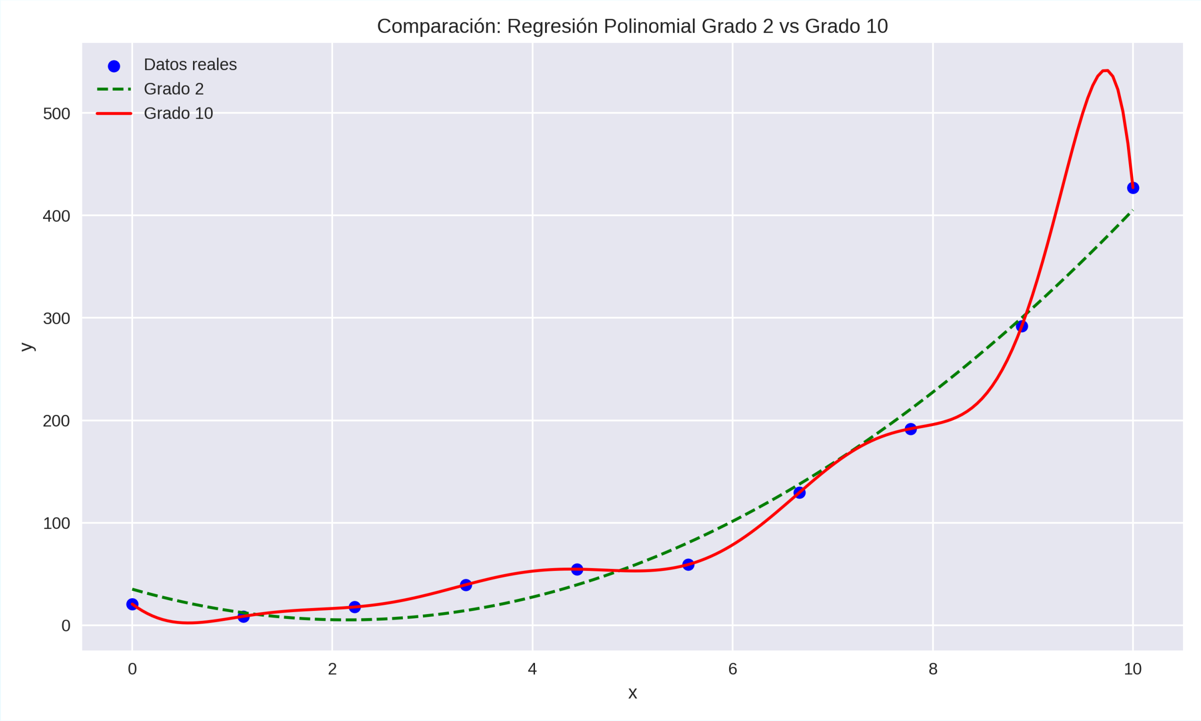
  $$y = \beta_0 + \beta_1 x$$

- **Grado 2**: Cuadrático

  $$y = \beta_0 + \beta_1 x + \beta_2 x^2$$

- **Grado 3**: Cúbico

  $$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$



Cuándo Usar

1. **Relaciones no lineales**: Cuando la relación no es lineal
2. **Curvatura**: Datos que muestran patrones curvos
3. **Puntos de inflexión**: Cuando hay cambios en la tendencia
4. **Ajuste mejorado**: Para mejorar el ajuste del modelo

Ejemplo: Predicción de Resistencia

Optimizar parámetros de fabricación para obtener una resistencia eléctrica específica.

Variables controladas:

x_1: Temperatura de cocción (°C)
x_2: Tiempo de cocción (minutos)
x_3: Espesor del material (µm)
x_4: Concentración de dopante (%)
y : Resistencia (Ω) — variable objetivo

Modelo de Regresión Múltiple

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \epsilon$$

Ejemplo interactivo

Fin