

Universidad Torcuato Di Tella

Trabajo Práctico 2

Métodos Computacionales

Optimización y Análisis de Datos

Grupo Labubu

Tomás Estrada, Agustín Rodríguez, Joaquín Naser

November 23, 2025

Índice

1	Introducción	2
2	Modelado y cuadrados mínimos	2
2.1	Sistema normal	2
2.2	Función en Python para resolver el sistema	3
2.3	Gráfico de datos y curva ajustada	3
2.4	Error cuadrático medio (MSE)	4
3	Interpretación mediante SVD	4
3.1	Autovalores de $A^\top A$ y valores singulares	4
3.2	Descomposición SVD y ortogonalidad de U	5
3.3	Solución por SVD y comparación con ecuaciones normales	5
3.4	Condicionamiento y colinealidad de las columnas	6
4	Análisis cuadrático del error	7
4.1	Visualización de $Q(\beta_1, \beta_2)$	7
4.2	Convexidad de $Q(\beta)$ y mínimo de mínimos cuadrados	8
5	Optimización numérica	9
5.1	Implementación del método	9
5.2	Experimentos con tasas de aprendizaje y tolerancias	9
6	Discusión y extensiones	10
6.1	Relación entre los distintos métodos	11
6.2	Modelo cúbico y efecto del ruido	11
7	Aplicación a un caso real	12
7.1	Construcción del modelo	13
7.2	Solución por ecuaciones normales	13
7.3	Solución mediante SVD	13
7.4	Evaluación del modelo	14
7.5	Visualización del ajuste	14
7.6	Conclusiones del caso real	15

1 Introducción

En numerosos contextos científicos y tecnológicos es necesario ajustar modelos a conjuntos de datos experimentales, estimar parámetros desconocidos y analizar la calidad de las soluciones obtenidas. En este trabajo práctico abordamos estas problemáticas desde el marco de la **optimización y el análisis de datos**, aplicando conceptos de *ortogonalidad, formas cuadráticas, descomposición en valores singulares (SVD)* y *métodos iterativos*.

El objetivo principal es comprender cómo estas herramientas se integran en un mismo proceso de modelización: desde la formulación matricial de un problema de mínimos cuadrados hasta la interpretación geométrica de sus soluciones y la evaluación del condicionamiento numérico. Además, se implementan distintas estrategias de resolución —*ecuaciones normales, SVD y descenso por gradiente*— para comparar su eficiencia y estabilidad.

Finalmente, los métodos desarrollados se aplican a un conjunto de datos reales con el fin de construir un modelo predictivo y analizar su desempeño, conectando la teoría del álgebra lineal con la práctica de la optimización numérica y el aprendizaje a partir de datos.

2 Modelado y cuadrados mínimos

Se dispone del conjunto de datos:

$$(x_i, y_i) = \{(0, 1), (1, 2), (2, 2.8), (3, 3.6), (4, 4.5)\}.$$

Queremos ajustar un modelo cuadrático de la forma:

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

mediante el método de los mínimos cuadrados.

2.1 Sistema normal

En forma matricial, definimos la matriz de diseño

$$A = \begin{bmatrix} 1 & 0 & 0^2 \\ 1 & 1 & 1^2 \\ 1 & 2 & 2^2 \\ 1 & 3 & 3^2 \\ 1 & 4 & 4^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ 2 \\ 2.8 \\ 3.6 \\ 4.5 \end{bmatrix}.$$

El estimador de mínimos cuadrados se obtiene resolviendo el sistema normal

$$A^\top A \beta = A^\top y.$$

2.2 Función en Python para resolver el sistema

Implementamos una función `gaussian_elimination_solve(M, b)` que resuelve sistemas lineales con eliminación gaussiana y pivoteo parcial, sin usar `np.linalg.solve`. Esta función se utiliza para resolver el sistema normal anterior y obtener los coeficientes $\hat{\beta}$ del modelo.

Aplicada a este problema se obtiene:

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} \approx \begin{bmatrix} 1.0314 \\ 0.9171 \\ -0.0143 \end{bmatrix}.$$

2.3 Gráfico de datos y curva ajustada

Con los coeficientes estimados, la curva ajustada queda dada por

$$\hat{y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2.$$

En la Figura 1 se observan los puntos originales y la curva cuadrática estimada. El modelo captura bastante bien la tendencia creciente y ligeramente cóncava de los datos.

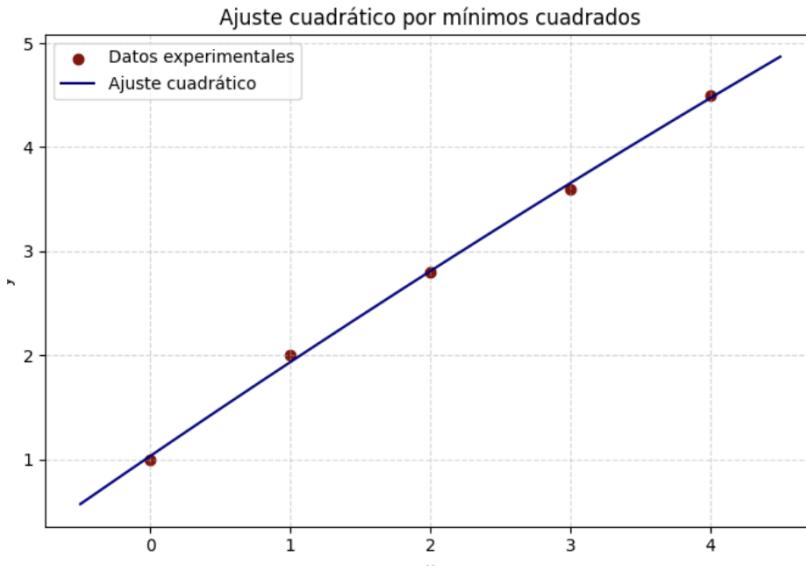


Figure 1: Ajuste cuadrático por mínimos cuadrados para los datos experimentales.

2.4 Error cuadrático medio (MSE)

El error cuadrático medio se calcula como:

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

Para los coeficientes obtenidos,

$$\text{MSE} \approx 1.83 \times 10^{-3},$$

lo que indica un ajuste muy bueno sobre este conjunto de 5 puntos.

3 Interpretación mediante SVD

En esta sección analizamos la matriz de diseño A mediante su descomposición en valores singulares (SVD). Esto permite:

- relacionar los autovalores de $A^\top A$ con los valores singulares de A ;
- verificar la ortogonalidad de las matrices de la SVD;
- obtener la solución de mínimos cuadrados usando SVD;
- estudiar el condicionamiento y la colinealidad de las columnas de A .

Recordemos que un sistema se dice *mal condicionado* cuando pequeñas perturbaciones en A o en y producen grandes variaciones en la solución $\hat{\beta}$. La SVD proporciona una forma natural de cuantificar esta sensibilidad.

3.1 Autovalores de $A^\top A$ y valores singulares

Partimos de la matriz

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix}.$$

En Python se calculó primero la matriz simétrica

$$A^\top A$$

y sus autovalores:

$$\lambda_1 \approx 385.00, \quad \lambda_2 \approx 3.47, \quad \lambda_3 \approx 0.52.$$

A partir de ellos se obtienen los valores singulares teóricos

$$\sigma_i = \sqrt{\lambda_i},$$

es decir:

$$\sigma_1 \approx 19.6215, \quad \sigma_2 \approx 1.8632, \quad \sigma_3 \approx 0.7237.$$

Luego, usando `np.linalg.svd(A)` se calcularon directamente los valores singulares de A . La diferencia entre estos y las raíces cuadradas de los autovalores de $A^\top A$ resultó del orden de

$$10^{-14},$$

lo que numéricamente confirma la relación teórica entre ambas cantidades.

3.2 Descomposición SVD y ortogonalidad de U

La SVD de A se escribe como

$$A = U\Sigma V^\top,$$

donde:

- U es una matriz 5×3 con columnas ortonormales (autovectores de AA^\top),
- Σ es diagonal con los valores singulares $(\sigma_1, \sigma_2, \sigma_3)$,
- V es una matriz 3×3 ortogonal (autovectores de $A^\top A$).

En el código se verificó la ortogonalidad de U calculando

$$\|U^\top U - I\|_\infty,$$

lo que dio

$$\|U^\top U - I\|_\infty \approx 9.53 \times 10^{-16}.$$

Este valor tan pequeño muestra que, dentro del error de redondeo de doble precisión, las columnas de U son efectivamente ortonormales.

3.3 Solución por SVD y comparación con ecuaciones normales

La solución de mínimos cuadrados se puede escribir en términos de la SVD como

$$\hat{\beta}_{\text{SVD}} = V\Sigma^{-1}U^\top y.$$

En Python se implementó esta fórmula usando las matrices U , S (vector de valores singulares) y V^\top devueltas por `np.linalg.svd`. Por otro lado, se había obtenido previ-

amente

$$\hat{\beta}_{\text{norm}}$$

resolviendo el sistema normal $A^\top A \beta = A^\top y$ con eliminación gaussiana.

Los resultados numéricos fueron:

$$\hat{\beta}_{\text{SVD}} = \hat{\beta}_{\text{norm}} \approx \begin{bmatrix} 1.0314 \\ 0.9171 \\ -0.0143 \end{bmatrix},$$

y la norma de la diferencia

$$\|\hat{\beta}_{\text{SVD}} - \hat{\beta}_{\text{norm}}\|_2 \approx 3.29 \times 10^{-15},$$

prácticamente cero. Esto confirma que ambas formas de resolver el problema de mínimos cuadrados son equivalentes en este caso.

3.4 Condicionamiento y colinealidad de las columnas

El número de condición en norma 2 de la matriz A se calcula como

$$\kappa_2(A) = \frac{\sigma_{\max}}{\sigma_{\min}} = \frac{19.6215}{0.7237} \approx 27.11.$$

Este valor indica que el problema no es perfectamente bien condicionado, pero tampoco está en una situación extrema: algunos errores de redondeo pueden amplificarse, aunque no de forma catastrófica.

Para estudiar la colinealidad entre columnas, se analizaron las columnas correspondientes a x y x^2 (previamente centradas) mediante:

- la matriz de correlación:

$$\text{corr} \approx \begin{bmatrix} 1 & 0.959 \\ 0.959 & 1 \end{bmatrix},$$

- el ángulo entre ambos vectores, que resultó

$$\theta \approx 16.48^\circ.$$

Una correlación tan alta y un ángulo pequeño sugieren que las columnas son bastante parecidas (alta colinealidad), lo cual se refleja en el número de condición moderadamente grande. Sin embargo, A sigue teniendo rango completo y el problema continúa siendo resoluble de manera estable.

4 Análisis cuadrático del error

La función de error del modelo se define como

$$Q(\beta) = \|A\beta - y\|^2,$$

y puede reescribirse como una forma cuadrática en los parámetros:

$$Q(\beta) = \beta^\top (A^\top A)\beta - 2(A^\top y)^\top \beta + y^\top y.$$

De esta expresión se identifican:

- **Término cuadrático:** $\beta^\top (A^\top A)\beta$.
- **Término lineal:** $2(A^\top y)^\top \beta$.
- **Término independiente:** $y^\top y$.

4.1 Visualización de $Q(\beta_1, \beta_2)$

Para estudiar la geometría de la función de error, fijamos el valor óptimo de β_0 y consideramos

$$Q(\beta_1, \beta_2) = Q(\beta_0^*, \beta_1, \beta_2)$$

sobre una grilla en el plano (β_1, β_2) . A partir de los valores calculados de Q se generó una superficie en 3D y sus curvas de nivel.

Las curvas de nivel muestran un paraboloide convexo cuyo mínimo global se alcanza en el punto

$$(\beta_1, \beta_2) = (\hat{\beta}_1, \hat{\beta}_2),$$

que coincide con la solución de mínimos cuadrados obtenida en la sección anterior. En el gráfico se marca este punto mínimo sobre el valle de la superficie (Figura 2).

Superficie de la función de error $Q(\beta_1, \beta_2)$ con β_0 fijo

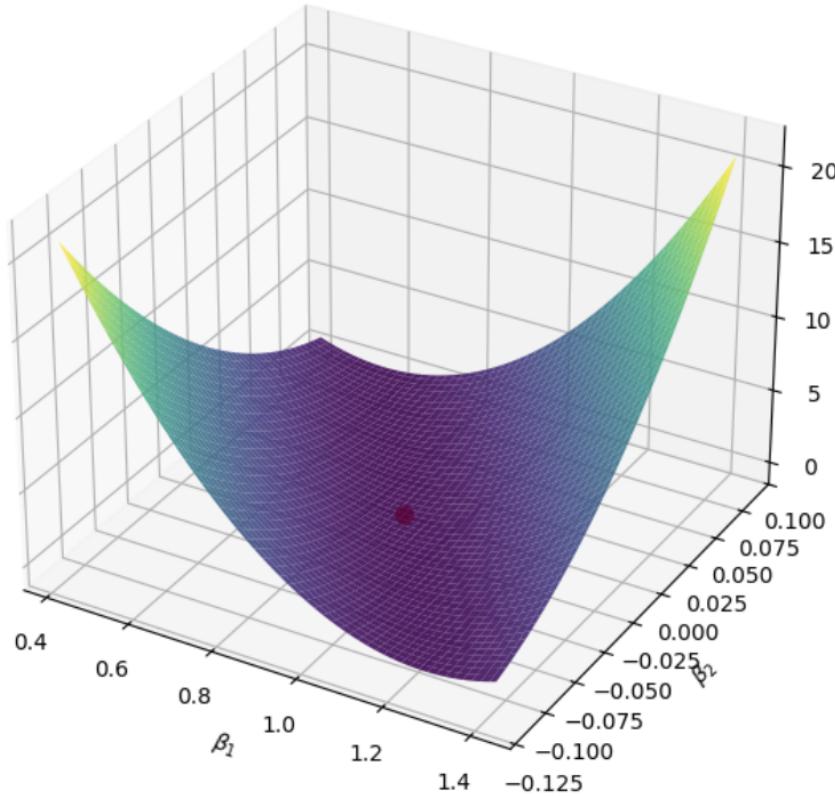


Figure 2: Superficie de la función de error $Q(\beta_1, \beta_2)$ con β_0 fijo.

4.2 Convexidad de $Q(\beta)$ y mínimo de mínimos cuadrados

A partir de la definición

$$Q(\beta) = \|A\beta - y\|^2$$

se obtiene el gradiente y la Hessiana:

$$\nabla Q(\beta) = 2A^\top(A\beta - y), \quad \nabla^2 Q(\beta) = 2A^\top A.$$

Como $A^\top A$ es una matriz simétrica y semidefinida positiva, la Hessiana $\nabla^2 Q(\beta)$ es también semidefinida positiva. Por lo tanto, $Q(\beta)$ es una función convexa y cualquier punto donde el gradiente se anule corresponde a un mínimo global.

Imponiendo la condición de estacionariedad,

$$\nabla Q(\beta) = 0 \implies A^\top A \hat{\beta} = A^\top y,$$

se recuperan exactamente las ecuaciones normales. Numéricamente se verificó que los autovalores de $2A^\top A$ son no negativos y que el gradiente evaluado en $\hat{\beta}$ es prácticamente

nulo, confirmando que $\hat{\beta}$ es el mínimo global de $Q(\beta)$.

5 Optimización numérica

En esta sección implementamos el método de *descenso por gradiente* para minimizar la función de error

$$Q(\beta) = \|A\beta - y\|^2,$$

cuyo gradiente está dado por

$$\nabla Q(\beta) = 2A^\top(A\beta - y).$$

Partimos de un valor inicial $\beta^{(0)}$ y actualizamos iterativamente

$$\beta^{(k+1)} = \beta^{(k)} - \eta \nabla Q(\beta^{(k)}),$$

donde $\eta > 0$ es la tasa de aprendizaje. El algoritmo se detiene cuando $\|\nabla Q(\beta^{(k)})\|_2 < \text{tol}$ o cuando se alcanza un número máximo de iteraciones.

5.1 Implementación del método

Se programó un bucle que, dados la matriz de diseño A , el vector de observaciones y y un valor inicial $\beta^{(0)} = 0$, realiza las siguientes operaciones en cada iteración:

1. Calcula el residuo $r^{(k)} = A\beta^{(k)} - y$.
2. Evalúa el gradiente $g^{(k)} = 2A^\top r^{(k)}$.
3. Actualiza los parámetros mediante $\beta^{(k+1)} = \beta^{(k)} - \eta g^{(k)}$.
4. Registra el valor de $Q(\beta^{(k)}) = \|r^{(k)}\|^2$.

Con una tasa de aprendizaje $\eta = 10^{-3}$ y una tolerancia exigente, el método requiere del orden de 10^4 – 10^5 iteraciones para acercarse a la solución de mínimos cuadrados. El vector final obtenido es numéricamente muy cercano a $\hat{\beta}$ (la distancia en norma 2 es del orden de 10^{-6}) y el valor de $Q(\beta)$ coincide con el mínimo hallado por las ecuaciones normales.

5.2 Experimentos con tasas de aprendizaje y tolerancias

El desempeño del descenso por gradiente depende fuertemente de la tasa de aprendizaje η y de la tolerancia del criterio de corte. En el notebook se probaron las combinaciones

$$\eta \in \{10^{-2}, 10^{-3}, 5 \times 10^{-4}\}, \quad \text{tol} \in \{10^{-6}, 10^{-8}\}.$$

Los resultados observados fueron:

- Para $\eta = 10^{-2}$ el método se vuelve inestable y las iteraciones producen valores numéricamente inválidos (NaN), lo que indica que la tasa de aprendizaje es demasiado grande.
- Para $\eta = 10^{-3}$ el algoritmo reduce rápidamente $Q(\beta)$ y se acerca al mínimo global en relativamente pocas iteraciones (comparado con tasas más chicas).
- Para $\eta = 5 \times 10^{-4}$ el método es estable pero avanza más lentamente: la función de costo decrece de manera más suave y, con el mismo número de iteraciones, la solución queda un poco más lejos de $\hat{\beta}$ que en el caso $\eta = 10^{-3}$.

En la Figura 3 se muestra, en escala semilogarítmica, la evolución de $\|\nabla Q(\beta^{(k)})\|_2$ para distintas tasas de aprendizaje: cuanto más rápido decrece la curva, más eficiente es la elección de η .

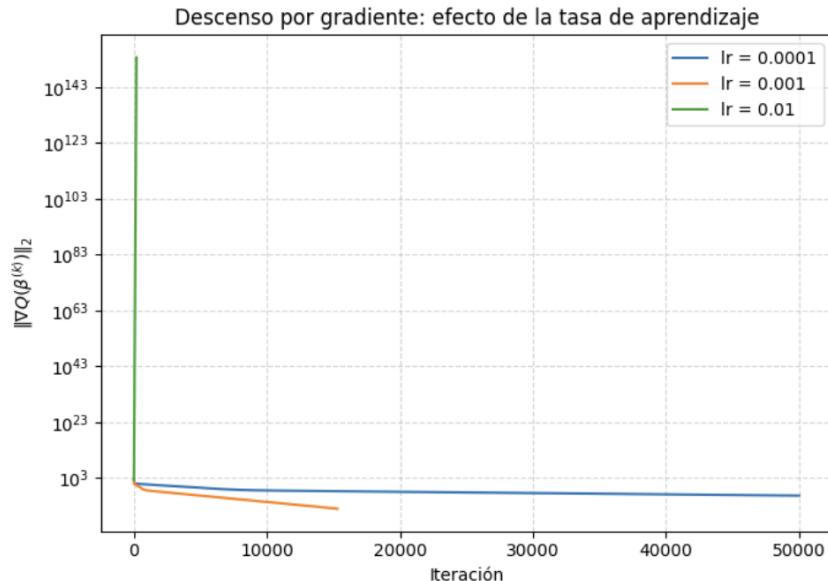


Figure 3: Descenso por gradiente: efecto de la tasa de aprendizaje sobre la convergencia.

En resumen, existe un compromiso entre estabilidad y velocidad de convergencia: valores de η demasiado grandes hacen divergir el algoritmo, mientras que valores demasiado pequeños lo vuelven innecesariamente lento.

6 Discusión y extensiones

En esta sección comparamos los distintos métodos utilizados (ecuaciones normales, SVD y descenso por gradiente) y analizamos cómo cambia el ajuste al incorporar un término cúbico y ruido a los datos.

6.1 Relación entre los distintos métodos

En el trabajo utilizamos tres enfoques para resolver el problema de mínimos cuadrados:

1. Ecuaciones normales

Resuelven directamente el sistema

$$A^\top A \beta = A^\top y.$$

Es el método más sencillo conceptualmente, pero puede volverse inestable si la matriz $A^\top A$ está mal condicionada.

2. Descomposición SVD

Escribimos

$$A = U\Sigma V^\top, \quad \hat{\beta}_{\text{SVD}} = V\Sigma^{-1}U^\top y.$$

Es el método más estable numéricamente y permite analizar el condicionamiento del problema mediante los valores singulares.

3. Descenso por gradiente

Minimiza la función de error

$$Q(\beta) = \|A\beta - y\|^2$$

mediante iteraciones del tipo

$$\beta_{k+1} = \beta_k - \eta \nabla Q(\beta_k).$$

Es útil cuando la dimensión del problema crece o cuando no conviene formar explícitamente $A^\top A$.

En nuestro problema de tamaño reducido, los tres métodos convergen a la misma solución de mínimos cuadrados. Las diferencias se observan en su estabilidad numérica, en el número de iteraciones (en el caso del descenso por gradiente) y en las consideraciones teóricas asociadas al condicionamiento de A .

6.2 Modelo cúbico y efecto del ruido

Extendimos el modelo original agregando un término cúbico:

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

Con cuatro parámetros y cinco puntos de datos, el modelo cúbico logra un ajuste casi perfecto sobre los datos originales, con un MSE aún menor que el modelo cuadrático.

En la Figura 4 se compara gráficamente el ajuste cuadrático y el cúbico sobre los mismos puntos.

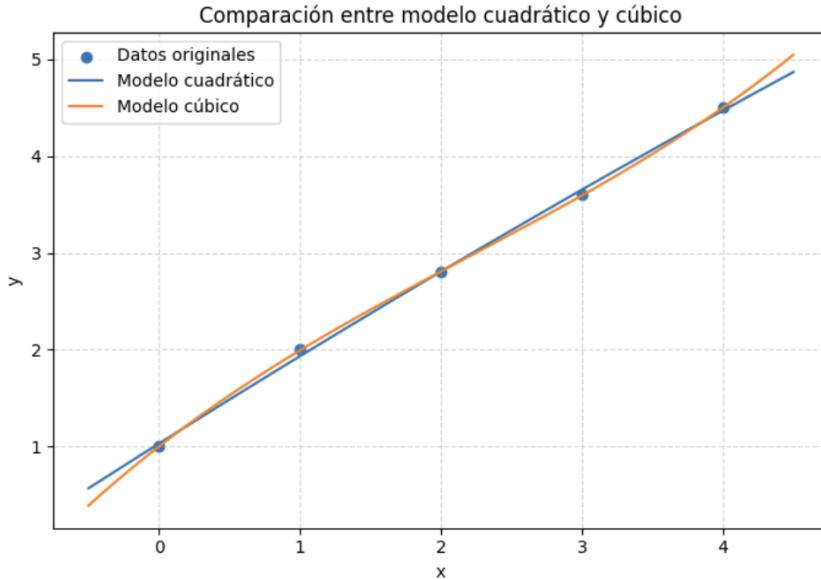


Figure 4: Comparación entre el modelo cuadrático y el modelo cúbico sobre los datos originales.

Sin embargo, este mayor poder de ajuste trae riesgos de *sobreajuste*, sobre todo cuando se agregan pocos datos o se introduce ruido. Para explorar este efecto, se añadió ruido gaussiano a las observaciones y se volvió a ajustar el modelo. Se observó que:

- el ajuste al conjunto de datos ruidoso sigue siendo bueno (el MSE respecto de los datos observados es pequeño);
- sin embargo, los coeficientes estimados cambian más respecto a los valores originales;
- en general, un modelo más complejo tiende a “perseguir” el ruido, y su capacidad de generalización fuera de la muestra puede empeorar.

Este ejercicio muestra cómo la complejidad del modelo y la calidad de los datos afectan directamente a la estabilidad del ajuste y a la interpretación de los parámetros estimados.

7 Aplicación a un caso real

Para ilustrar el uso de los métodos estudiados, se aplicó un modelo de regresión lineal a un dataset real proveniente del archivo **Salary Data.csv**. El objetivo fue predecir el salario anual de una persona a partir de su cantidad de años de experiencia.

7.1 Construcción del modelo

A partir del conjunto de datos se seleccionó la variable respuesta $y = \text{Salary}$ y la variable predictora $X = \text{Years of Experience}$. Esta elección se justifica porque:

- permite construir un modelo lineal simple y directamente interpretable;
- facilita el análisis matricial y la comparación entre métodos (ecuaciones normales y SVD);
- y permite aislar la relación base entre experiencia laboral y salario.

La matriz de diseño utilizada fue:

$$A = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix},$$

donde x_i representa los años de experiencia y y_i el salario correspondiente.

7.2 Solución por ecuaciones normales

La solución de mínimos cuadrados se obtuvo resolviendo

$$A^\top A \hat{\beta} = A^\top y$$

mediante la misma rutina de eliminación gaussiana desarrollada en la primera parte del trabajo. Esto produce un vector

$$\hat{\beta}_{\text{norm}} = (\beta_0, \beta_1),$$

donde β_0 es el intercepto (salario esperado con experiencia nula) y β_1 mide el incremento promedio de salario por cada año adicional de experiencia.

7.3 Solución mediante SVD

En forma análoga, se calculó la descomposición $A = U\Sigma V^\top$ y la solución

$$\hat{\beta}_{\text{SVD}} = V\Sigma^{-1}U^\top y.$$

En el notebook se verificó que ambas soluciones coinciden dentro del error numérico esperado, lo cual confirma la consistencia y estabilidad del problema para este conjunto de datos.

7.4 Evaluación del modelo

La predicción se obtuvo como $\hat{y} = A\hat{\beta}$ y el error cuadrático medio se definió como

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

Además del MSE, se analizó el histograma de residuos del modelo y la nube de puntos de *salario real vs salario predicho*. Los residuos se distribuyen aproximadamente alrededor de cero y presentan una forma unimodal, lo que indica que el modelo captura de manera razonable la tendencia general, aunque persisten variaciones no explicadas que probablemente dependen de factores adicionales (nivel educativo, puesto laboral, industria, etc.).

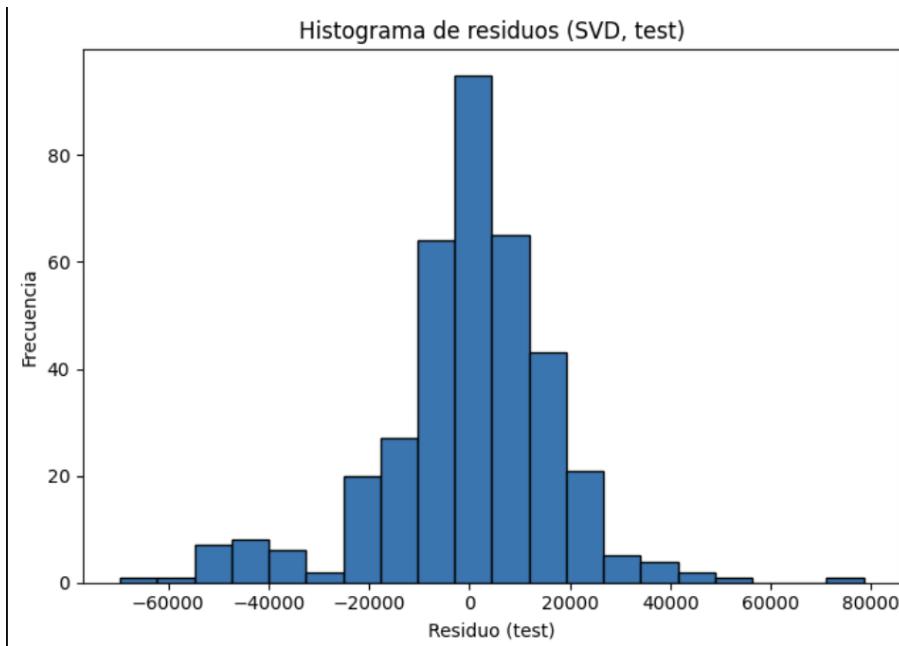


Figure 5: Histograma de residuos del modelo lineal ajustado al dataset Salary Data.

7.5 Visualización del ajuste

En la Figura 6 se muestra la recta ajustada junto con los datos reales, donde se observa una relación creciente entre experiencia y salario: a mayor cantidad de años de experiencia, mayor salario esperado.

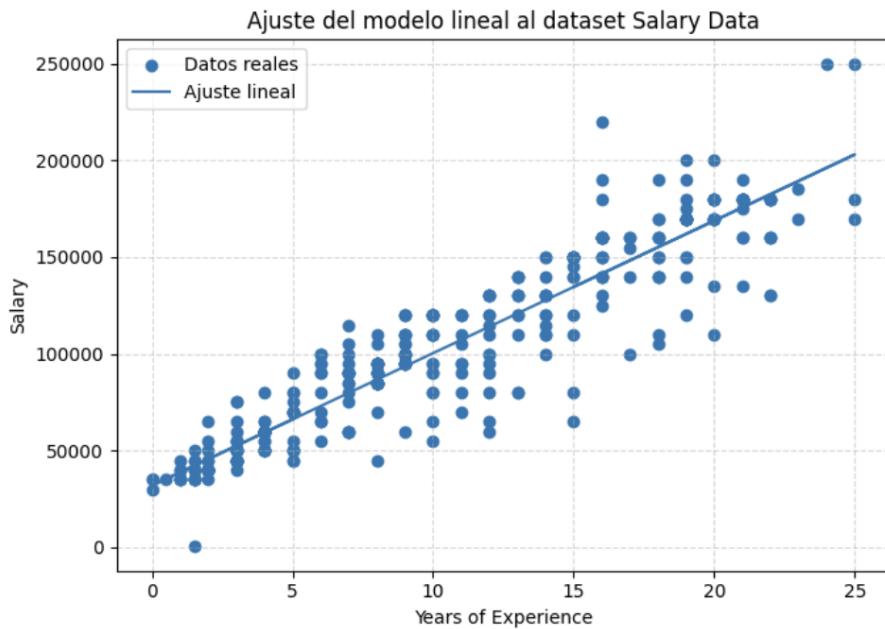


Figure 6: Ajuste del modelo lineal al dataset Salary Data (Salary vs Years of Experience).

7.6 Conclusiones del caso real

En síntesis, el modelo lineal simple logra capturar de forma adecuada la relación creciente entre los años de experiencia y el salario anual. Las soluciones obtenidas mediante ecuaciones normales y SVD coincidieron numéricamente, lo que confirma la estabilidad del problema. No obstante, el análisis del MSE y de los residuos sugiere que existen factores adicionales que influyen en el salario. La incorporación de nuevas variables o técnicas de regularización podría mejorar el rendimiento del modelo en futuras extensiones.